intel

MCS-48™ FAMILY OF SINGLE CHIP MICROCOMPUTERS

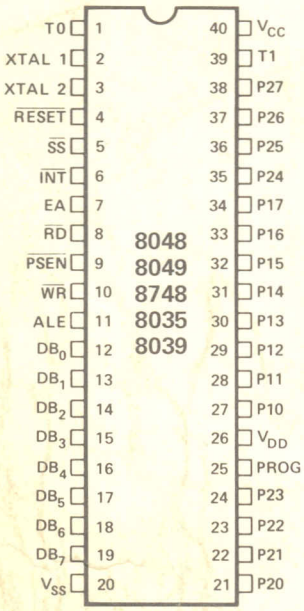
USER'S MANUAL

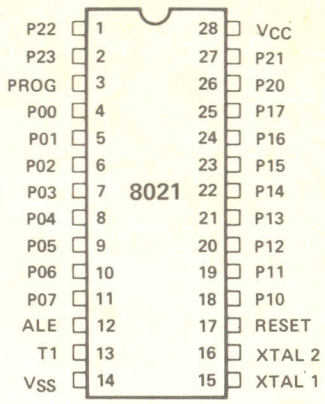


uly 1978
©Intel Corporation, 1977, 1978

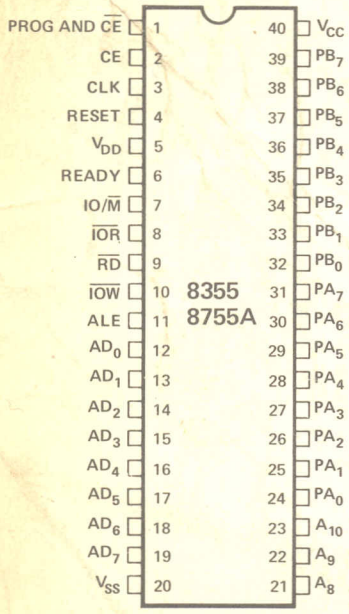
9800270D
$5.00

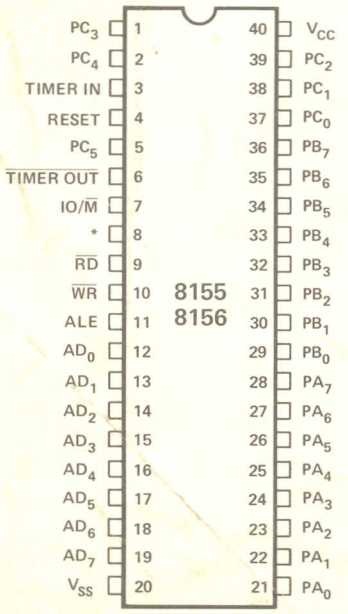
MCS-48™ Pin Configurations



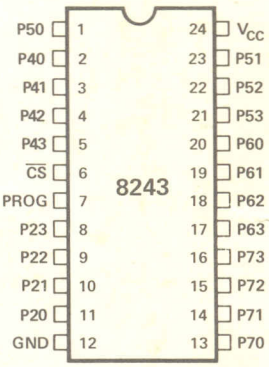


quantum electronics
Box 391262
Bramley
2018





*8155 = $\overline{CE}$
8156 = CE



©INTEL CORPORATION, 1978

TABLE 1. REGISTER ACCUMULATOR.

R_r	MOV A,R	MOV R,A	XCH A,R	MOV R, #DATA	INC R	DEC R	DJNZ R	ADD A,R	ADDC A,R	ANL A,R	ORL A,R	XRL A,R
R0	F8	A8	28	B8	18	C8	E8	68	78	58	48	D8
R1	F9	A9	29	B9	19	C9	E9	69	79	59	49	D9
R2	FA	AA	2A	BA	1A	CA	EA	6A	7A	5A	4A	DA
R3	FB	AB	2B	BB	1B	CB	EB	6B	7B	5B	4B	DB
R4	FC	AC	2C	BC	1C	CC	EC	6C	7C	5C	4C	DC
R5	FD	AD	2D	BD	1D	CD	ED	6D	7D	5D	4D	DD
R6	FE	AE	2E	BE	1E	CE	EE	6E	7E	5E	4E	DE
R7	FF	AF	2F	BF	1F	CF	EF	6F	7F	5F	4F	DF

TABLE 2. INPUT/OUTPUT.

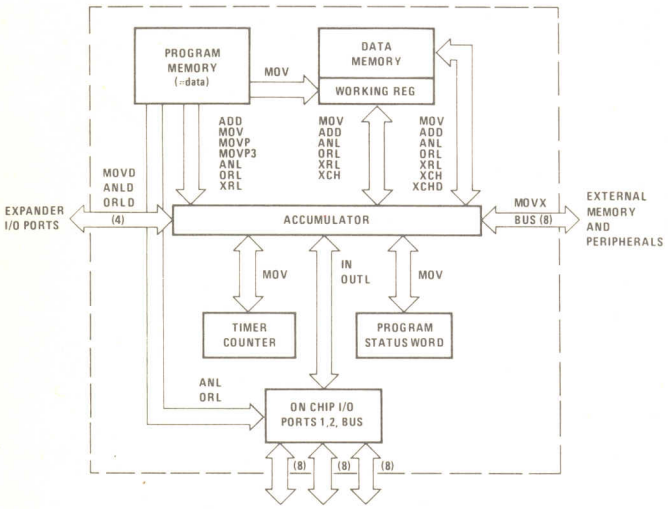
Port	IN	OUT	AND	OR
BUS	08	02	98	88
P1	09	39	99	89
P2	0A	3A	9A	8A
P4	0C	3C	9C	8C
P5	0D	3D	9D	8D
P6	0E	3E	9E	8E
P7	0F	3F	9F	8F

TABLE 3. BRANCH.

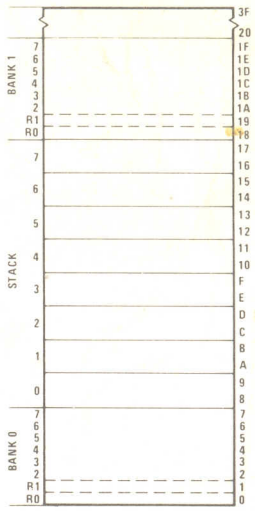
Page	JMP	CALL
0	04	14
1	24	34
2	44	54
3	64	74
4	84	94
5	A4	B4
6	C4	D4
7	E4	F4

Page = 256 bytes

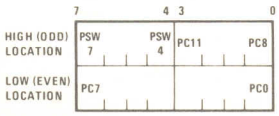
MCS-48™ DATA TRANSFER INSTRUCTIONS



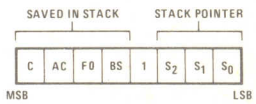
DATA RAM



STACK FORMAT



PROGRAM STATUS WORD (PSW)



C CARRY
AC AUXILIARY CARRY
F0 FLAG 0
BS REGISTER BANK SELECT

1978

MCS-48™ MICROCOMPUTER USER'S MANUAL

Intellec and MCS are Registered Trademarks of Intel Corporation

Copyright ©1978 by Intel Corporation. All rights reserved, no part of this publication including any mnemonics contained herein may be reproduced without the prior written permission of Intel Corporation, 3065 Bowers Avenue, Santa Clara, CA 95051

TABLE OF CONTENTS

Chapter 1 Introduction

1.0 Introduction to MCS-48™ 1-1

1.1 The Functions of a Computer 1-5
- Typical Computer System 1-5
- Architecture of a CPU 1-5
- Computer Operations 1-8

1.2 Programming a Microcomputer 1-10
- Machine Language Programming 1-10
- Assembly Language Programming 1-11

1.3 Developing an MCS-48™ Based Product 1-13
- Education 1-13
- Function Definition 1-13
- Hardware Configuration 1-13
- Code Generation 1-13
- PROMPT 48 1-14
- Intellec Development System 1-15
- Production 1-15

Chapter 2 The Single Component MCS-48™ System

8048/8748/8035 and 8049/8039

2.0 Summary 2-1

2.1 Architecture 2-1

2.2 Pin Description 2-14

2.3 Programming, Verifying and Erasing EPROM 2-16

2.4 Test and Debug 2-18

8021

2.5 Program Memory 2-21

2.6 Data Memory 2-21

2.7 Oscillator and Clock 2-22

2.8 Timer/Event Counter 2-22

2.9 Input/Output Capabilities 2-23

2.10 CPU 2-25

2.11 Reset 2-25

Chapter 3 The Expanded MCS-48™ System

3.0 Summary 3-1

3.1 Expansion of Program Memory 3-1
- Instruction Fetch Cycle 3-1
- Extended Addressing 3-1
- Restoring Port Information 3-2
- Expansion Examples 3-2

3.2 Expansion of Data Memory 3-4
- Read Write Cycle 3-4
- Addressing External Memory 3-5
- Examples of Data Memory Expansion 3-5

Chapter 3 Continued

3.3 Expansion of Input/Output 3-5
I/O Expander Device 3-6
Expansion with Standard Peripherals 3-7
Combination Memory and I/O Expanders 3-7
Expansion Examples 3-8
3.4 Multi-Chip MCS-48™ Systems 3-9
3.5 Bank Switching 3-10
3.6 Control Signal Summary 3-11
3.7 Port Characteristics 3-11

Chapter 4 Instruction Set

4.0 Introduction 4-1
Data Transfers 4-1
Accumulator Operations 4-2
Register Operations 4-2
Flags 4-2
Branch Instructions 4-2
Subroutines 4-3
Timer Instructions 4-3
Control 4-3
Input/Output Instructions 4-4
4.1 Instruction Set Description 4-4
Summary 4-5
Symbols and Abbreviations 4-6
Alphabetic Listing by Mnemonic 4-7

Chapter 5 Application Examples

5.0 Introduction 5-1
5.1 Hardware Examples 5-1
Frequency Reference Options 5-1
Stand Alone 8048/8049 5-2
Stand Alone 8021 5-3
Multiple Interrupt Sources 5-4
Multiple Prioritized Interrupts 5-5
External Program Memory 5-6
Program Memory and I/O Expansion 5-7
Data Memory and I/O Expansion 5-8
A Three Chip System 5-9
I/O Expansion Techniques 5-10
8049 Emulator Circuit 5-16
General Application Examples 5-18
5.2 Software Examples 5-21

Chapter 6 MCS-48™ Component Specifications

8048 ROM Microcomputer 6-1
8648 EPROM Microcomputer 6-1
8748 EPROM Microcomputers 6-1
8035 Microcomputers 6-1
8049/8039/ 8039-6 Microcomputers 6-9
8021 Microcomputer 6-15
8022 Microcomputer 6-21
8355 ROM and I/O Expander 6-33
8755 EPROM and I/O Expander 6-39
8155/56 RAM and I/O Expander 6-45
8243 MCS-48™ I/O Expander 6-57

Chapter 7 Compatible MCS-80™ Components

8308 8192 Bit Static MOS RAM 7-1
2316E 16,384 Bit Static MOS ROM 7-5
8708 8192 1K x 8 EPROM 7-9
2716 16K UV Erasable PROM 7-13
8101A-4 1024 Bit Static MOS RAM With Separate I/O 7-17
8111A-4 1024 Bit Static MOS RAM With Common I/O 7-21
5101 1024 Bit Static CMOS RAM 7-25
8212 Eight-Bit Input/Output Port 7-29
8255A Programmable Peripheral Interface 7-39
8251 Programmable Communication Interface 7-61
8205 High Speed 1 Out of 8 Binary Decoder 7-77
8214 Priority Interrupt Control Unit 7-83
8216/8226 4-Bit Parallel Bi-Directional Bus Driver 7-87
8253 Programmable Interval Timer 7-93
8259 Programmable Interrupt Controller 7-105
8279 Programmable Keyboard Interface 7-121
8278 Programmable Keyboard Interface 7-133
8041/8741 Universal Peripheral Interface 8-Bit Microcomputers 7-143

Chapter 8 Support Products

Intellec® Prompt 48™ MCS-48™ Microcomputer Design Aid 8-1
Intellec® Microcomputer Development System 8-7
MCS-48™ Diskette-Based Software Support Package 8-12
ICE-48 MCS-48 In-Circuit Emulator 8-14
MDS-EM1 8021 Emulation Board 8-19
MCS-48™ System Workshop 8-22

Appendices

Packing Information A1-1
Ordering Information A2-1

Chapter 1
INTRODUCTION



INTRODUCTION

1.0 Introduction to MCS-48™ 1-1
1.1 Functions of a Computer 1-5
1.2 Programming a Microcomputer 1-10
1.3 Developing An MCS-48™ Based Product 1-13

- **The 8049/8039 is now 80% faster! See page 6-9.**
- **The 8022 is an 8021 with more Memory, I/O, and an A/D converter. See page 6-21.**

INTRODUCTION

1.0 Introduction to MCS-48™

Recent advances in NMOS technology have allowed Intel for the first time to place enough capability on a single silicon die to create a true single-chip microcomputer containing all the functions required in a digital processing system. A set of such microcomputers on single chips, their variations, and optional peripherals are collectively called the MCS-48 microcomputer family. These products are fully described in this manual.

The head of the family is the 8048 microcomputer which contains the following functions in a single 40 pin package:

8-Bit CPU
1K x 8 ROM Program Memory
64 x 8 RAM Data Memory
27 I/O Lines
8-Bit Timer/Event Counter

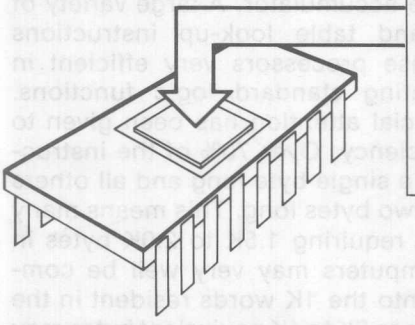
A 2.5 or 5.0 microsecond cycle time and a repertoire of over 90 instructions each consisting of either one or two cycles makes the single chip 8048 the equal in performance of most presently available multi-chip NMOS microprocessors. The 8048 is, however, a true "low-cost" microcomputer. A single 5V supply requirement for all MCS-48 components assures that "low cost" also applies to the power supply in your system.

New Family Members

The MCS-48 family of microcomputers which began with the 8048 and 8748 has now been expanded with new members which provide either more capability or lower cost than the original family members. While broadening the applications possible with a single chip microcomputer, these new microcomputers share both the instruction set and development support of the 8048.

The 8049 is a single-chip microcomputer which is completely interchangeable with the 8048, but contains twice the program memory and twice the data memory of the 8048. The 8035 and 8039 are compatible processors without internal program memory. The 8039 contains twice the data memory of the 8035.

The 8021 is a new very low cost MCS-48 family member which contains a subset of



8049	8048	8021	FEATURES
✓	✓	✓	8 BIT CPU
2K x 8	1K x 8	1K x 8	PROGRAM MEMORY
128 x 8	64 x 8	64 x 8	DATA RAM
27	27	21	I/O LINES
✓	✓	✓	TIMER COUNTER
✓	✓	✓	OSCILLATOR AND CLOCK
✓	✓	✓	RESET CIRCUIT
✓	✓		INTERRUPT

ON CHIP FEATURES

the 8048's instruction set and incorporates several new features critical in low cost applications.

Even with low component costs; however, a project may be jeopardized by high development and rework costs resulting from an inflexible production design. Intel has solved this problem by creating two pin-compatible versions of the 8048 microcomputer: the 8048 with mask Programmable ROM program memory for low cost production and the 8748 with user programmable and erasable EPROM program memory for prototype development. The 8748 is essentially a single chip microcomputer "breadboard" which can be modified over and over again during development and pre-production then replaced by the low cost 8021*, 8048, or 8049 ROM for volume production. The 8748 provides a very easy transition from development to production and also provides an easy vehicle for temporary field updates while new ROMs are being made.

SPECIAL FEATURES

- **SINGLE 5V SUPPLY**
- **40 PIN DIP OR 28 PIN DIP**
- **PIN COMPATIBLE ROM AND EPROM**
- **2.5, 5.0 AND 10.0 μsec CYCLE VERSIONS**
- **ALL INSTRUCTIONS 1 OR 2 CYCLES**
- **SINGLE STEP**
- **8 LEVEL STACK**
- **2 WORKING REGISTER BANKS**
- **RC, LC, XTAL, OR EXTERNAL FREQUENCY SOURCE**
- **OPTIONAL CLOCK OUTPUT**
- **POWER DOWN STANDBY MODE**

To allow the MCS-48 to solve a wide range of problems and to provide for future expansion, all 8048 and 8049 functions have been made externally expandable using either special expanders or standard memories and peripherals. An efficient low cost means of I/O expansion is provided by either the 8243 I/O Expander or standard TTL or CMOS circuits. The 8243 provides 16 I/O lines in a 24 pin package. For systems with large I/O requirements, multiple 8243s can be used.

For such applications as Keyboards, Displays, Serial communication lines, etc. standard MCS-80/85 peripheral circuits may be added. Program and data memory may be expanded using standard memories or the 8355 and 8155 memories that also include programmable I/O lines and timing functions.

For applications which require a more custom tailored interface, the 8041 or 8741 Universal Peripheral Interface (UPI-41) devices can be used. The UPI-41 devices are available in both ROM and EPROM versions and are essentially slave versions of the 8048/8748 which are designed to interface directly with expandable MCS-48 processors and provide flexible intelligent I/O capability. The 8041/8741 share the instruction set of the MCS-48 family of processors.

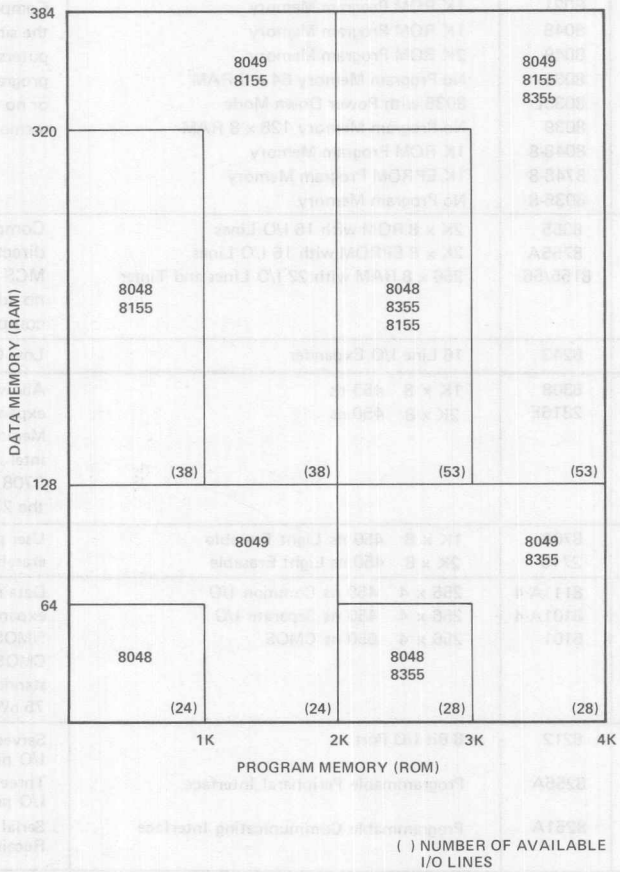
The 8035 and 8039 are an 8048 or 8049 respectively without internal program memory that allows the user to match his program memory requirements exactly by using a wide variety of external memories. The 8035 and 8039 allow the user to select a minimum cost system no matter what his program memory requirements. The 8035L is an 8035 with the powerdown mode of the 8048.

The MCS-48 processors are designed to be efficient control processors as well as arithmetic processors. They provide an instruction set which allows the user to directly set and reset individual lines within its I/O ports as well as test individual bits within the accumulator. A large variety of branch and table look-up instructions make these processors very efficient in implementing standard logic functions. Also, special attention has been given to code efficiency. Over 70% of the instructions are a single byte long and all others are only two bytes long. This means many functions requiring 1.5K to 2.0K bytes in other computers may very well be compressed into the 1K words resident in the 8048 or up to 3K to 4K equivalent bytes may be compressed into the 8049.

*The 8021 is code compatible but not pin compatible with the 8748.

	FUNCTION	PART NUMBER	DESCRIPTION	COMMENTS
MCS-48™	Microcomputers	8021 8048 8049 8035 8035L 8039 8048-8 8748-8 8035-8	1K ROM Program Memory 1K ROM Program Memory 2K ROM Program Memory No Program Memory 64 x 8 RAM 8035 with Power Down Mode No Program Memory 128 x 8 RAM 1K ROM Program Memory 1K EPROM Program Memory No Program Memory	Compatible versions of the single chip microcomputers provide mask programmed, light erasable, or no internal program memory.
	Memory and I/O Expanders	8355 8755A 8155/56	2K x 8 ROM with 16 I/O Lines 2K x 8 EPROM with 16 I/O Lines 256 x 8 RAM with 22 I/O Lines and Timer	Compatible devices allow direct expansion of MCS-48 functions with no additional external components.
	I/O Expander	8243	16 Line I/O Expander	Low Cost I/O Exapnder
Compatible MCS-80/85™ Components	Standard ROMs	8308 2316E	1K x 8 450 ns 2K x 8 450 ns	Allow low cost external expansion of Program Memory. The 8308 is interchangeable with 8708 and the 2316E with the 2716.
	Standard EPROM	8708 2716	1K x 8 450 ns Light Erasable 2K x 8 450 ns Light Erasable	User programmable and erasable.
	Standard RAMs	8111A-4 8101A-4 5101	256 x 4 450 ns Common I/O 256 x 4 450 ns Separate I/O 256 x 4 650 ns CMOS	Data memory can be easily expanded using standard NMOS RAMs. The 5101 CMOS equivalent reduces standby power to 75 nW/bit
	Standard I/O	8212 8255A 8251A	8-Bit I/O Port Programmable Peripheral Interface Programmable Communicating Interface	Serves as Address Latch or I/O port. Three 8-bit programmable I/O ports. Serial Communications Receiver/Transmitter
	Standard Peripherals	8205 8214 8216 8226 8253 8259 8279 8278	1 of 8 Binary Decoder Priority Interrupt Controller Bi-directional Bus Driver Bi-directional Bus Driver (Inverting) Programmable Interval Timer Programmable Interrupt Controller Programmable Keyboard/Display Interface (64 Keys) Programmable Keyboard/Display Interface (128 Keys)	MCS-80 peripheral devices are compatible with the MCS-48 allowing easy addition of such specialized interfaces as the 8279 Keyboard/Display Interface. Future MCS-80/85 devices will also be compatible.
	Universal Peripheral Interface	8041 8741	ROM Program Memory EPROM Program Memory	User programmable to perform any custom I/O and control functions.

MCS-48™ MICROCOMPUTER COMPONENTS



THE EXPANDED MCS-48™ SYSTEM

The chart above shows the expansion possibilities using the 8048 and 8049 in various combinations with the Intel® 8355/8755 Program Memory and I/O Expander and the 8155 Data Memory and I/O Expander. Data Memory can be expanded beyond the resident words in blocks of 256 by adding 8155's. Program Memory can be expanded beyond the resident 1K or 2K in blocks of 2K by using the 8355/8755 in combination with the 8048 or 8049. If all external memory is desired, the 8035 or 8039 can be substituted for the 8048 and 8049.

1.1 The Function of a Computer

This chapter introduces certain basic computer concepts. It provides background information and definitions which will be useful in later chapters of this manual. Those already familiar with computers may skip this material, at their option.

1.1.1 A Typical Computer System

A typical digital computer consists of:

A central processor unit (CPU)
Program Memory
Data Memory
Input/output (I/O) ports

The processor memory serves as a place to store Instructions, the coded pieces of information that direct the activities of the CPU, while Memory stores the Data, the coded pieces of information that are processed by the CPU. A group of logically related instructions stored in memory is referred to as a Program. The CPU "reads" each instruction from memory in a logically determined sequence, and uses it to initiate processing actions. If the program sequence is coherent and logical, processing the program will produce intelligible and useful results. The program must be organized such that the CPU does not read a non-instruction word when it expects to see an instruction.

The CPU can rapidly access any data stored in memory; but often the memory is not large enough to store the entire data bank required for a particular application. The problem can be resolved by providing the computer with one or more Input Ports. The CPU can address these ports and input the data contained there. The addition of input ports enables the computer to receive information from external equipment (such as a paper tape reader or floppy disk) at high rates of speed and in large volumes.

A computer also requires one or more Output Ports that permit the CPU to communicate the result of its processing to the outside world. The output may go to a display, for use by a human operator, to a peripheral device that produces "hard-copy", such as a line-printer, to a peripheral storage device, such as a floppy disk unit, or the output may constitute process control signals that direct the operations of another system, such as an automated assembly line. Like input ports, output ports are addressable. The input and output ports together permit the processor to communicate with the outside world.

The CPU unifies the system. It controls the functions performed by the other components. The CPU must be able to fetch instructions from memory, decode their binary contents and execute them. It must also be able to reference memory and I/O ports as necessary in the execution of instructions. In addition, the CPU should be able to recognize and respond to certain external control signals, such as INTERRUPT requests. The functional units within a CPU that enable it to perform these functions are described below.

1.1.2 The Architecture of a CPU

A typical central processor unit (CPU) consists of the following interconnected functional units:

Registers
Arithmetic/Logic Unit (ALU)
Control Circuitry

Registers are temporary storage units within the CPU. Some registers, such as the program counter and instruction register, have dedicated uses. Other registers, such as the accumulator, are for more general purpose use.

Accumulator

The accumulator usually stores one of the operands to be manipulated by the ALU. A typical instruction might direct the ALU to add the contents of some other register to the contents of the accumulator and store the result in the accumulator itself. In general, the accumulator is both a source (operand) and a destination (result) register. Often a CPU will include a number of additional general purpose registers that can be used to store operands or intermediate data. The availability of general purpose registers

eliminates the need to "shuffle" intermediate results back and forth between memory and the accumulator, thus improving processing speed and efficiency.

Program Counter (Jumps, Subroutines and the Stack):

The instructions that make up a program are stored in the system's memory. The central processor references the contents of memory in order to determine what action is appropriate. This means that the processor must know which location contains the next instruction.

Each of the locations in memory is numbered, to distinguish it from all other locations in memory. The number which identifies a memory location is called its Address. The processor maintains a counter which contains the address of the next program instruction. This register is called the Program Counter. The processor updates the program counter by adding "1" to the counter each time it fetches an instruction, so that the program counter is always current (pointing to the next instruction).

The programmer therefore stores his instructions in numerically adjacent addresses, so that the lower addresses contain the first instructions to be executed and the higher addresses contain later instructions. The only time the programmer may violate this sequential rule is when an instruction in one section of memory is a Jump instruction to another section of memory.

A jump instruction contains the address of the instruction which is to follow it. The next instruction may be stored in any memory location, as long as the programmed jump specifies the correct address. During the execution of a jump instruction, the processor replaces the contents of its program counter with the address embodied in the Jump. Thus, the logical continuity of the program is maintained.

A special kind of program jump occurs when the stored program "Calls" a subroutine. In this kind of jump, the processor is required to "remember" the contents of the program counter at the time that the jump occurs. This enables the processor to resume execution of the main program when it is finished with the last instruction of the subroutine.

A Subroutine is a program within a program. Usually it is a general-purpose set of instructions that must be executed repeatedly in the course of a main program. Routines which calculate the square, the sine, or the logarithm of a program variable are good examples of functions often written as subroutines. Other examples might be programs designed for inputting data to a particular peripheral device.

The processor has a special way of handling subroutines, in order to insure an orderly return to the main program. When the processor receives a Call instruction, it increments the Program Counter and stores the counter's contents in a reserved memory area known as the Stack. The Stack thus saves the address of the instruction to be executed after the subroutine is completed. Then the processor loads the address specified in the Call into its Program Counter. The next instruction fetched will therefore be the first step of the subroutine.

The last instruction in any subroutine is a Return. Such an instruction need specify no address. When the processor fetches a Return instruction, it simply replaces the current contents of the Program Counter with the address on the top of the stack. This causes the processor to resume execution of the calling program at the point immediately following the original Call instruction.

Subroutines are often Nested; that is, one subroutine will sometimes call a second subroutine. The second may call a third, and so on. This is perfectly acceptable, as long as the processor has enough capacity to store the necessary return addresses, and the logical provision for doing so. In other words, the maximum depth of nesting is determined by the depth of the stack itself. If the stack has space for storing three return addresses, then

three levels of subroutines may be accommodated.

Instruction Register and Decoder

Every computer has a Word Length that is characteristic of that machine. A computer's word length is usually determined by the size of its internal storage elements and interconnecting paths (referred to as Buses); for example, a computer whose registers and buses can store and transfer 8-bits of information has a characteristic word length of 8-bits and is referred to as an 8-bit parallel processor. An 8-bit parallel processor generally finds it most efficient to deal with 8-bit binary fields, and the memory associated with such a processor is therefore organized to store 8-bits in each addressable memory location. Data and instructions are stored in memory as 8-bit binary numbers, or as numbers that are integral multiples of 8-bits: 16-bits, 24-bits, and so on. This characteristic 8-bit field is often referred to as a Byte. If however, efficient handling of 4 or even 1-bit data is necessary special processor instructions can provide this capability.

Each operation that the processor can perform is identified by a unique byte of data known as an Instruction Code or Operation Code. An 8-bit word used as an instruction code can distinguish between 256 alternative actions, more than adequate for most processors.

The processor fetches an instruction in two distinct operations. First, the processor transmits the address in its Program Counter to the program memory. Then the program memory returns the addressed byte to the processor. The CPU stores this instruction byte in a register known as the Instruction Register, and uses it to direct activities during the remainder of the instruction execution.

The 8-bits stored in the instruction register can be decoded and used to selectively activate one of a number of output lines. Each line represents a set of activities associated with execution of a particular instruction code. The enabled line can be combined with selected timing pulses, to develop electrical signals that can then be used to initiate specific actions. This translation of code into action is performed by the Instruction Decoder and by the associated control circuitry.

An 8-bit instruction code is often sufficient to specify a particular processing action. There are times, however, when execution of the instruction requires more information than 8-bits can convey.

One example of this is when the instruction references a memory location. The basic instruction code identifies the operation to be performed, but cannot specify the object address as well. In a case like this, a two byte instruction must be used. Successive instruction bytes are stored in sequentially adjacent memory locations, and the processor performs two fetches in succession to obtain the full instruction. The first byte retrieved from memory is placed in the processor's instruction register, and subsequent byte is placed in temporary storage; the processor then proceeds with the execution phase.

Address Register(s)

A CPU may use a register to hold the address of a memory location that is to be accessed for data. If the address register is Programmable, (i.e., if there are instructions that allow the programmer to alter the contents of the register) the program can "build" an address in the address register prior to executing a Memory Reference instruction (i.e., an instruction that reads data from memory, writes data to memory or operates on data stored in memory).

Arithmetic/Logic Unit (ALU)

All processors contain an arithmetic/logic unit, which is often referred to simply as the ALU. The ALU, as its name implies, is that portion of the CPU hardware which performs the arithmetic and logical operations on the binary data.

The ALU must contain an Adder which is capable of combining the contents of two registers in accordance with the logic of binary arithmetic. This provision permits the

processor to perform arithmetic manipulations on the data it obtains from memory and from its other inputs.

Using only the basic adder a capable programmer can write routines which will subtract, multiply and divide, giving the machine complete arithmetic capabilities. In practice, however, most ALUs provide other built-in functions, including boolean logic operations, and shift capabilities.

The ALU contains Flag Bits which specify certain conditions that arise in the course of arithmetic and logical manipulations. It is possible to program jumps which are conditionally dependent on the status of one or more flags. Thus, for example, the program may be designed to jump to a special routine if the carry bit is set following an additional instruction.

Control Circuitry

The control circuitry is the primary functional unit within a CPU. Using clock inputs, the control circuitry maintains the proper sequence of events required for any processing task. After an instruction is fetched and decoded, the control circuitry issues the appropriate signals (to units both internal and external to the CPU) for initiating the proper processing action. Often the control circuitry will be capable of responding to external signals, such as an interrupt. An Interrupt request will cause the control circuitry to temporarily interrupt main program execution, jump to a special routine to service the interrupting device, then automatically return to the main program.

1.1.3 Computer Operations

There are certain operations that are basic to almost any computer. A sound understanding of these basic operations is a necessary prerequisite to examining the specific operations of a particular computer.

Timing

The activities of the central processor are cyclical. The processor fetches an instruction, performs the operations required, fetches the next instruction, and so on. This orderly sequence of events requires precise timing, and the CPU therefore requires a free running oscillator clock which furnishes the reference for all processor actions. The combined fetch and execution of a single instruction is referred to as an Instruction Cycle. The portion of a cycle identified with a clearly defined activity is called a State. And the interval between pulses of the timing oscillator is referred to as a Clock Period. As a general rule, one or more clock periods are necessary for the completion of a state, and there are several states in a cycle.

Instruction Fetch

The first state(s) of any instruction cycle will be dedicated to fetching the next instruction. The CPU issues a read signal and the contents of the program counter are sent to program memory, which responds by returning the next instruction word. The first byte of the instruction is placed in the instruction register. If the instruction consists of more than one byte, additional states are required to fetch the second byte of the instruction. When the entire instruction is present in the CPU, the program counter is incremented (in preparation for the next instruction fetch) and the instruction is decoded. The operation specified in the instruction will be executed in the remaining states of the instruction cycle. The instruction may call for a data memory read or write, an input or output and/or an internal CPU operation, such as a register-to-register transfer or an add operation.

Memory Read

An instruction fetch is merely a special program memory read operation that brings the instruction to the CPU's instruction register. The instruction fetched may then call for data to be read from data memory into the CPU. The CPU again issues a read signal and sends the proper memory address; memory responds by returning the requested word. The data received is placed in the accumulator or one of the other general purpose registers (not the instruction register).

Memory Write

A memory write operation is similar to a read except for the direction of data flow. The CPU issues a write signal, sends the proper memory address, then sends the data word to be written into the addressed data memory location.

Input/Output

Input and Output operations are similar to memory read and write operations with the exception that an I/O port is addressed instead of a memory location. The CPU issues the appropriate input or output control signal, sends the proper address and either receives the data being input or sends the data to be output.

Data can be input/output in either parallel or serial form. All data within a digital computer is represented in binary coded form. A binary data word consists of a group of bits; each bit is either a one or a zero. Parallel I/O consists of transferring all bits in the word at the same time, one bit per line. Serial I/O consists of transferring one bit at a time on a single line. Naturally serial I/O is much slower, but it requires considerable less hardware than does parallel I/O.

Interrupts

Interrupt provisions are included on many central processors, as a means of improving the processor's efficiency. Consider the case of a computer that is processing a large volume of data, portions of which are to be output to a printer. The CPU can output a byte of data within a single machine cycle but it may take the printer the equivalent of many machine cycles to actually print the character specified by the data byte. The CPU could then remain idle waiting until the printer can accept the next data byte. If an interrupt capability is implemented on the computer, the CPU can output a data byte then return to data processing. When the printer is ready to accept the next data byte, it can request an interrupt. When the CPU acknowledges the interrupt, it suspends main program execution and automatically branches to a routine that will output the next data byte. After the byte is output, the CPU continues with main program execution. Note that this is, in principle, quite similar to a subroutine call, except that the jump is initiated externally rather than by the program.

More complex interrupt structures are possible, in which several interrupting devices share the same processor but have different priority levels. Interruptive processing is an important feature that enables maximum utilization of a processor's capacity for high system throughput.

1.2 Programming a Microcomputer

1.2.1 Machine Language Programming

A microprocessor is instructed what to do by programming it with a series of instructions stored in Program Memory. The processor fetches these instructions one at a time and performs the operation indicated. These instructions must be stored in a form that the processor can understand. This format is referred to as Machine Language. For most microprocessors this instruction is a group of 8 binary bits (1's and 0's) called a word (also called a byte if the word is 8-bits). Some instructions require more than one location in Program Memory. To execute a multi-byte instruction, the processor must execute multiple fetches of program memory before performing the instruction. Because multi-byte instructions take more Program Memory and take longer to execute than single byte instructions their use is usually kept to a minimum.

A processor may be programmed by writing a sequence of instructions in the binary code (ones and zeros) which the machine can interpret directly. This is machine language programming and it is very useful where the program to be written is small and the application requires that the designer have an intimate knowledge of the microprocessor. Machine language programming allows the user, because of his detailed knowledge, to use many programming "tricks" to produce the most compact and efficient code possible.

The following is an example of a machine language program: This program reads 5 sequential 8-bit words in from an I/O port and stores them sequentially in data memory. The program starts by initializing two registers, one which determines where the data is to be stored and another which counts the number of words to be stored. When finished the processor continues on to the next instructions.

Step Number	Machine Code	Explanation
0	1011 1000	Load decimal 32 in register R0
1	0010 0000	
2	1011 1010	Load decimal 5 in register R2
3	0000 0101	
4	0000 1001	Load Port 1 to accumulator
5	1111 0000	Transfer contents of accumulator to register addressed by register 0
6	0001 1000	Increment R0 by 1
7	1110 1010	Decrement register 2 by 1, if result is zero continue to step 9, if not go to step 4
8	0000 0100	
9	—	
10	—	

As you can see, writing machine instructions in ones and zeros can be very laborious and subject to error. It is almost always more efficient to represent each 8-bits if machine language code in a shorthand format called Hexadecimal. The term hexadecimal results from the character set used in hexadecimal notation. Hexadecimal is merely an extension of the normal decimal numbers by the addition of the first six letters of the alphabet. This gives a total of 16 different characters. Each hexadecimal "digit" can represent 16 values or the equivalent of four binary bits; therefore, each 8-bit machine language word can be represented by 2 hexadecimal (hex for short) digits. The correspondence among the decimal, binary, and hex number systems is given below:

Decimal	Hex	Binary
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Our machine language program then becomes:

Step	Hex Code
0	B8
1	20
2	BA
3	05
4	09
5	F0
6	18
7	EA
8	04

This coding is now quite efficient to write and read and coding errors are much easier to detect. Hex coding is usually very efficient for small programs (a few hundred lines of code) however, it does have two major limitations in larger programs:

1. Hex coding is not self-documenting, that is, the code itself does not give any indication in human terms of the operation to be performed. The user must learn each code or constantly use a Program Reference Card to convert.

2. Hex coding is absolute, that is, the program will work only when stored in a specific location in program memory. This is because the branch or jump instructions in the program reference specific addresses elsewhere in the program. In the example above steps 7 and 8 reference step (or address) 4. If the program were to be moved, step 8 would have to be changed to refer to the new address of step 4.

1.2.2 Assembly Language Programming

Assembly language overcomes the disadvantages of machine language by allowing the use of alphanumeric symbols to represent machine operation codes, branch addresses, and other operands. For example, the instruction to increment the contents of register 0 becomes INC R0 instead of the hex 18, giving the user at a glance the meaning of the instruction. Our example program can be written in assembly language as follows:

Step No.	Hex Code		Assembly Code
0	B8		MOV R0, #32
1	20		
2	BA		MOV R2, #05
3	05		
4	09	INP:	IN A, P1
5	F0		MOV @R0, A
6	18		INC R0
7	EA		DJNZ R2, INP
8	04		

The first statement can be verbalized as follows: Move to Register 0 the decimal number 32. Move instructions are always structured such that the destination is first and the source is second. The pound sign "#" indicates that the source is "immediate" data (data contained in the following byte of program memory). In this case data was specified as a decimal 32, however, this could have been written as a hex 20H or a binary 0010 0000B since the assembler will accept either form. Notice also that in this instance two lines of hex code are represented by one line of assembly code.

The input instruction IN A, P1 has the same form as a MOV instruction indicating that the contents of Port 1 are to be transferred to the accumulator. In front of the input instruction is an address lable which is delineated by a colon. This lable allows the program to be written in a form independent of its final location in program memory since the branch instruction at the end of the program can refer to this lable rather than a specific address. This is a very important advantage of assembly language programs since it

allows instructions to be added or deleted throughout the program during debugging without requiring that any jump addresses be changed.

The next instruction MOV @R0, A can be verbalized as, Move to the data memory location addressed by R0, the contents of the accumulator. The @ sign indicates an indirect operation whereby the contents of either register 0 or register 1 acts as a pointer to the data memory location to be operated on.

The last instruction is a Decrement and Jump if Not Zero instruction which acts in combination with the specified register as a loop counter. In this case register 2 is loaded with 5 initially and then decremented by one each time the loop is executed. If the result of the decrement is not zero, the program jumps to INP and executes another input operation. The fifth time thru the loop the result is zero and execution falls through to whatever routine follows the DJNZ instruction.

In addition to the normal features provided by assemblers, more advanced assemblers such as that for the MCS-48 offer such things as evaluation of expressions at assembly time, conditional assembly, and macro capability.

1. Evaluation of Expressions - Certain assemblers allow the use of arithmetic expressions and multiple symbols in the operand portion of instructions. For instance the MCS-48 assembler accepts instructions such as:

```
ADD A, # ALFA*BETA/2
```

ALFA and BETA are two previously defined symbols. At assembly time the expression ALFA*BETA/2 will be evaluated and the resulting number (which is the average of ALFA and BETA) will be treated as immediate data and designated as the second byte of the ADD immediate instruction. This expression has allowed the immediate data of this instruction to be defined in a single statement and eliminated the need for a third symbol equal to ALFA*BETA/2.

2. Conditional Assembly - Conditional assembly allows the programmer to select only certain portions of his assembly language (source) program for conversion to machine (object) code at assembly time. This allows for instance, the inclusion of various "debug" routines to be included in the program during development. Using conditional assembly, they can then be left out when the final assembly is done.

Conditional assembly also allows several versions of one basic program to be generated by selecting various portions of a larger program at assembly time.

3. Macro's - A macro instruction is essentially a symbol which is recognized by the assembler to represent a specific sequence of several standard instructions. A macro is a shorthand way of generating the same sequence of instructions at several locations in a program without having to rewrite the sequence each time it is used. For example, a typical macro instruction might be one which performs a subtract operation. The 8048 does not have a subtract instruction as such but the operation can be performed easily with three instructions:

```
CPL A
ADD A, REG
CPL A
```

This routine subtracts a register from the accumulator and leaves the result in the accumulator. This sequence can be defined as a macro with the name SUB and an operand which can be R0 to R7. To subtract R7 from the accumulator then, the programmer merely has to write:

```
SUB R7
```

and the assembler will automatically insert the three instructions above with R7 substituted for REG.

Once the assembly language source code is written it can be converted to machine executable object code by passing it through an assembler program. The MCS-48 assembler is a program which runs on the 8080-based Intellec MDS system explained in the next section.

1.3 Developing An MCS-48™ Based Product

Although the development of a microcomputer based product may differ in detail from the development cycle of a product based on TTL logic or relays, the basic procedures are the same — only the tools are different.

1.3.1 Education

The first step of course is to become familiar with what the microcomputer is and what it can do. The first step in this education is this document, the MCS-48™ User's Manual. The user's manual gives a detailed description of the MCS-48 family of components and how they may be used in various system configurations. Also included is a description of the 8048 instruction set and examples of how the instructions may be used. For a more complete discussion of the instruction set and programming techniques the MCS-48 Assembly Language Manual is also available.

If time is critical in getting started in microcomputers, individuals can attend one of many Intel sponsored 3-day training courses which give basic instruction in the MCS-48 as well as hands-on experience with MCS-48 development systems. These courses are a convenient means of getting started with the MCS-48, particularly for those not familiar with microprocessors.

After general familiarization is complete, either through self-instruction or a training course, the next step is to gain a better "feel" for what a microprocessor can do in your own applications by writing several exercise programs which perform basic functions. You may require such things as I/O routines, delays, counting functions, look-up tables, arithmetic functions, and logical operations which can serve as a set of building blocks for future applications programs. Several basic programming examples are included in the MCS-48 Assembly Language Manual while the Intel User's Library is a source of more specific applications routines.

1.3.2 Function Definition

After a thorough understanding of the microprocessor is achieved, the functions to be implemented can be defined using a flowchart method to describe each basic system function and the sequence in which the processor executes these functions. Once the system is flowcharted, critical time-related functions can be identified and sample programs written to verify that performance requirements can be met.

1.3.3 Hardware Configuration

The next step involves the definition of the microcomputer hardware required to implement the function. Input/Output capability must be defined in terms of number of inputs, number of outputs, bi-directional lines, latching or non-latching I/O, output drive capability, and input impedance. The number of words of RAM storage required for intermediate results and data storage must then be determined. The type of system will dictate whether battery backup is needed to maintain data RAM during power failure.

Probably the most difficult parameter to define initially is the amount of program memory needed to store the applications program. Although previously written exercise programs will make this estimate more accurate, a generous amount of "breathing room" should be allowed in program memory until coding is complete and the exact requirements are known. Many special functions such as serial communications (TTY) or keyboard/display interfaces may be implemented in software (programs); however, in cases where these functions place a severe load on the processor in terms of time or program memory, special peripheral interface circuits such as the 8251, Universal Synchronous or Asychronous Receiver/Transmitter (USART) or 8279 Keyboard/Display interface may be used.

1.3.4 Code Generation

The writing of the final program code for the application can begin once the system function and hardware have been defined and can be generated in parallel with the detailed hardware design (PC card layout, power supply, etc.)

At this point, there are two paths available to the designer/programmer and two types of design development aids provided by Intel to simplify the procedures. One system, called PROMPT 48, is a low cost development system which supports machine language programming and the second is the Intellec Microcomputer Development System which supports both machine and assembly languages. For those of you unfamiliar with the advantages and disadvantages of machine and assembly languages see Section 1.2.

1.3.5 PROMPT 48

PROMPT 48 is a low cost design aid consisting of: an 8748 processor to execute programs, control circuitry to provide debug functions such as single step and break points, a monitor program stored in ROM, an EPROM programmer, and a hexadecimal keyboard and display. There are two processor sockets on the front of PROMPT 48, one for programming the 8748 and one in which a programmed 8748 executes its program while under control of the monitor routine.

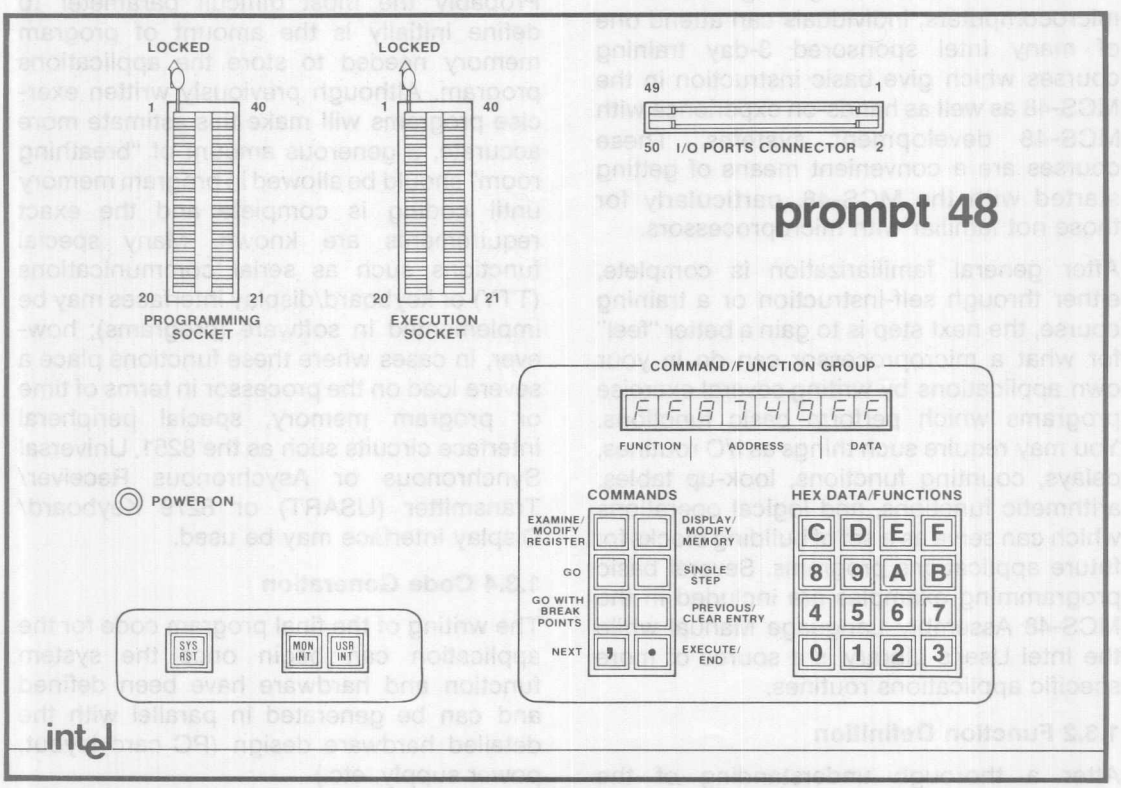
Use of PROMPT 48 involves the following steps:

1. Loading an application program into the PROMPT RAM memory via Hex keyboard or external terminal (TTY and RS232 interface provided).

2. Inserting an erased 8748 in the programming socket and transferring the application program to its internal EPROM.

3. Transferring programmed 8748 to execution socket where program is executed and debugged under control of the monitor.

The monitor routine allows the user to single step this processor, examine or modify all internal registers and data memory; or to run at full speed and stop the processor at predetermined breakpoints. PROMPT 48



also provides 1K of writeable program memory which may be used to debug user programs. A multiple single step feature is also provided in which the processor steps through its program dumping all internal contents to external RAM where it may be later displayed or typed out on an external terminal. Paper tape input and output in Intel's hexadecimal format is also available through the TTY.

1.3.6 Intellec Development System

The Intellec Microcomputer Development System is a modular development system which can be expanded as necessary to meet the requirements of your design cycle. The system consists of the processor unit which is based on Intel's 8080A microprocessor, and several optional units such as the UPP Universal PROM Programmer, the PTR High Speed Paper tape reader, the DOS Disk Operating System, and the Intellec CRT terminal.

To support the development of MCS-48 systems a macro-assembler ASM 48 is available for the Intellec System as well as a personality module for the UPP which will program the EPROM of the 8748. Also to be provided is in-circuit emulation capability with ICE-48 which will allow emulation and debug of user's 8048 application programs on the 8080A-based Intellec Development System.

The Intellec system is a flexible high performance development system which can support Intel's various microcomputer families with various optional modules. The macro-assembler and text editor programs provided allow the designer to write and edit his programs in assembly language and then generate the machine language output necessary to program the 8748 EPROM. The availability of a high speed CRT and a diskette operating system eliminates the laborious input and output of paper tape files normally required during the assembly process. Finally, ICE 48 allows the user to extend the resources of his entire Intellec system into the 8048 socket of his own system and use all its emulation, debug, and display facilities directly.

1.3.7 Production

Once a working program has been achieved, a preproduction phase usually follows where several prototype systems are evaluated in simulated situations or in actual operation in the field. During this period the use of the 8748 EPROM allows quick alteration of the application program when problems or suggested changes arise. Depending on the magnitude and number of future changes anticipated, the first production units may also be shipped with EPROM processor. However, to achieve the maximum cost reduction potential in high volume applications, a conversion to the 8048 ROM is usually necessary. This is an easy transition since the 8048 and 8748 are pin and machine code compatible equivalents. The user merely develops a hexadecimal tape of his 8748 program memory contents using his Intellec System or PROMPT 48 development aid and sends it to Intel along with his 8048 order. As the 8048 ROM's arrive they can immediately replace the 8748 EPROMs.

Chapter 2

THE SINGLE COMPONENT MCS-48™ SYSTEM



THE SINGLE COMPONENT MCS-48™ SYSTEM

SECTION 1: 8048/8748/8035 and 8049/8039

2.0 Summary 2-1
2.1 Architecture 2-1
2.2 Pin Description 2-14
2.3 Programming, Verifying and Erasing EPROM 2-16
2.4 Test and Debug 2-18

SECTION 2: 8021

2.5 Program Memory 2-21
2.6 Data Memory 2-21
2.7 Oscillator and Clock 2-22
2.8 Timer/Event Counter 2-22
2.9 Input/Output Capabilities 2-23
2.10 CPU 2-25
2.11 Reset 2-25

THE SINGLE COMPONENT MCS-48™ SYSTEM

2.0 Summary

Sections 2.1 through 2.4 describe in detail the functional characteristics of the 8748 EPROM, 8048/8049 ROM and 8035/8039 single component microcomputers. Unless otherwise noted, details within these sections apply to all versions. Sections 2.5 through 2.11 describe the operation of the 8021. This chapter is limited to those functions useful in single-chip implementations of the MCS-48. Chapter 3 discusses functions which allow expansion of program memory, data memory, and input-output capability.

2.1 Architecture

The following sections break the 8048 into functional blocks and describe each in detail.

2.1.1 Arithmetic Section

The arithmetic section of the processor contains the basic data manipulation functions of the 8048 and can be divided into the following blocks:

Arithmetic Logic Unit (ALU)
Accumulator
Carry Flag
Instruction Decoder

In a typical operation data stored in the accumulator is combined in the ALU with data from another source on the internal bus (such as a register or I/O port) and the result is stored in the accumulator or another register. The following is a more detailed description of the function of each block:

Instruction Decoder

The operation code (op code) portion of each program instruction is stored in the Instruction Decoder and converted to outputs which control the function of each of the blocks of the Arithmetic Section. These lines control the source of data and the destination register as well as the function performed in the ALU.

Arithmetic Logic Unit

The ALU accepts 8-bit data words from one or two sources and generates an 8-bit result under control of the Instruction Decoder. The ALU can perform the following functions:

Add With or Without Carry
And, OR, Exclusive OR
Increment/Decrement
Bit Complement
Rotate Left, Right
Swap Nibbles
BCD Decimal Adjust

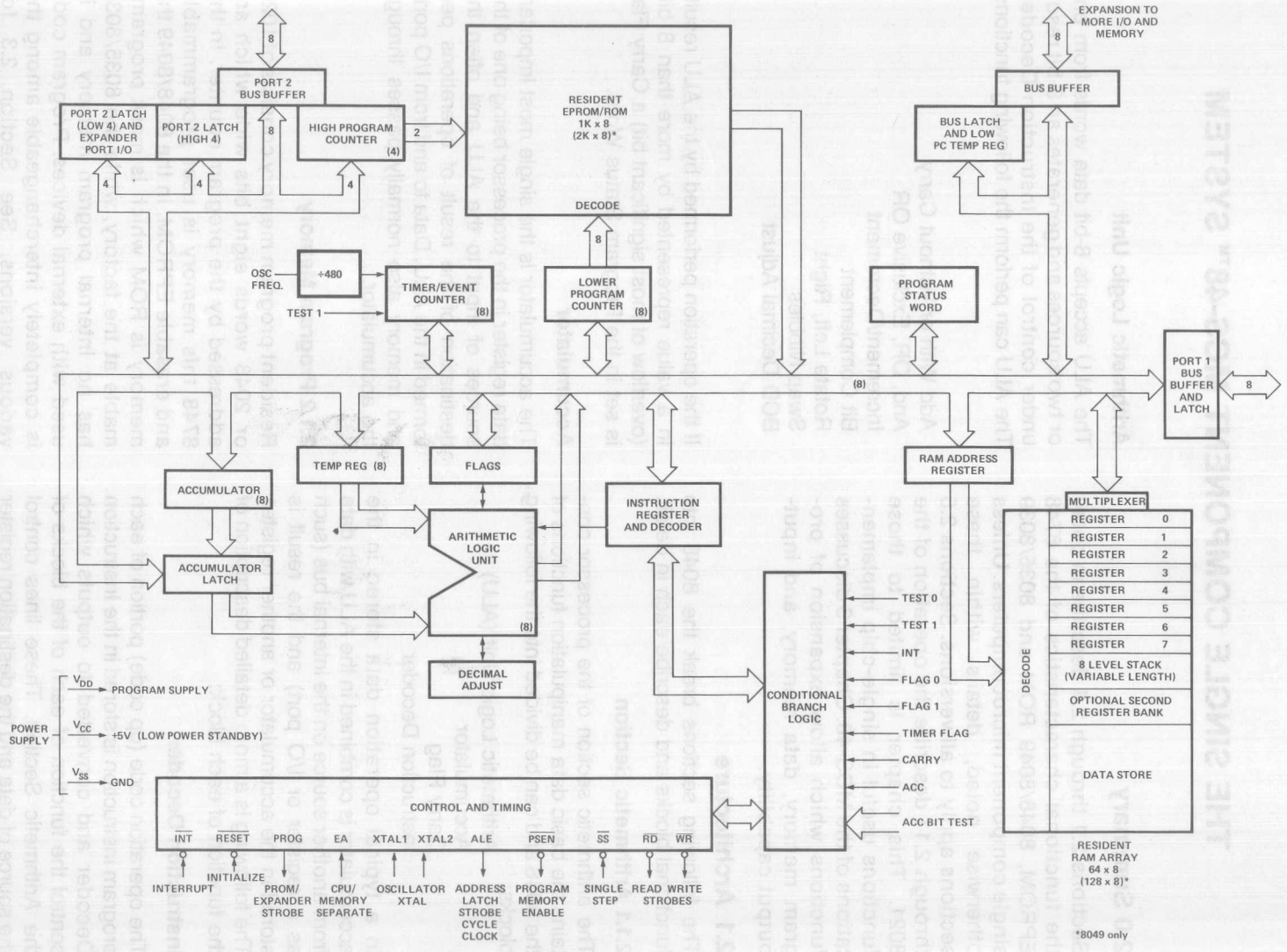
If the operation performed by the ALU results in a value represented by more than 8 bits (overflow of most significant bit) a Carry Flag is set in the Program Status Word.

Accumulator

The accumulator is the single most important data register in the processor being one of the sources of input to the ALU and often the destination of the result of operations performed in the ALU. Data to and from I/O ports and memory also normally passes through the accumulator.

2.1.2 Program Memory

Resident program memory consists of 1024 or 2048 words eight bits wide which are addressed by the program counter. In the 8748 this memory is user programmable and erasable EPROM, in the 8048/8049 the memory is ROM which is mask programmable at the factory, while the 8035/8039 has no internal program memory and is used with external devices. Program code is completely interchangeable among the various versions. See Section 2.3 for EPROM programming techniques.


8
PORT 2 BUS BUFFER
PORT 2 LATCH (LOW 4) AND EXPANDER PORT I/O
PORT 2 LATCH (HIGH 4)
HIGH PROGRAM COUNTER
(4)
2
RESIDENT EPROM/ROM 1K x 8 (2K x 8)*
DECODE
EXPANSION TO MORE I/O AND MEMORY
BUS BUFFER
BUS LATCH AND LOW PC TEMP REG
OSC FREQ.
÷480
TIMER/EVENT COUNTER
(8)
TEST 1
LOWER PROGRAM COUNTER
PROGRAM STATUS WORD
PORT 1 BUS BUFFER AND LATCH
TEMP REG (8)
FLAGS
RAM ADDRESS REGISTER
ACCUMULATOR
ACCUMULATOR LATCH
ARITHMETIC LOGIC UNIT
INSTRUCTION REGISTER AND DECODER
MULTIPLEXER
REGISTER 0
REGISTER 1
REGISTER 2
REGISTER 3
REGISTER 4
REGISTER 5
REGISTER 6
REGISTER 7
8 LEVEL STACK (VARIABLE LENGTH)
OPTIONAL SECOND REGISTER BANK
DATA STORE
DECODE
DECIMAL ADJUST
CONDITIONAL BRANCH LOGIC
TEST 0
TEST 1
INT
FLAG 0
FLAG 1
TIMER FLAG
CARRY
ACC
ACC BIT TEST
POWER SUPPLY
V_{DD} PROGRAM SUPPLY
V_{CC} +5V (LOW POWER STANDBY)
V_{SS} GND
CONTROL AND TIMING
INT
RESET
PROG
EA
XTAL1
XTAL2
ALE
PSEN
SS
RD
WR
INTERRUPT
INITIALIZE
PROM/ EXPANDER STROBE
CPU/ MEMORY SEPARATE
OSCILLATOR XTAL
ADDRESS LATCH STROBE CYCLE CLOCK
PROGRAM MEMORY ENABLE
SINGLE STEP
READ STROBE
WRITE STROBES
RESIDENT RAM ARRAY 64 x 8 (128 x 8)*
*8049 only

There are three locations in Program Memory of special importance:

LOCATION 0

Activating the Reset line of the processor causes the first instruction to be fetched from location 0.

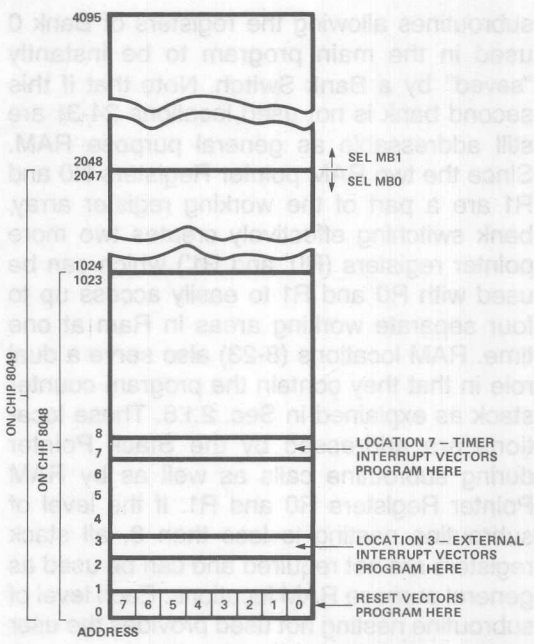
LOCATION 3

Activating the Interrupt input line of the processor (if interrupt is enabled) causes a jump to subroutine.

LOCATION 7

A timer/counter interrupt resulting from timer/counter overflow (if enabled) causes a jump to subroutine.

Therefore, the first instruction to be executed after initialization is stored in location 0, the first word of an external interrupt service subroutine is stored in location 3, and the first word of a timer/counter service routine is stored in location 7. Program memory can be used to store constants as well as program instructions. Instructions such as MOVP and MOVP3 allow easy access to data "lookup" tables.

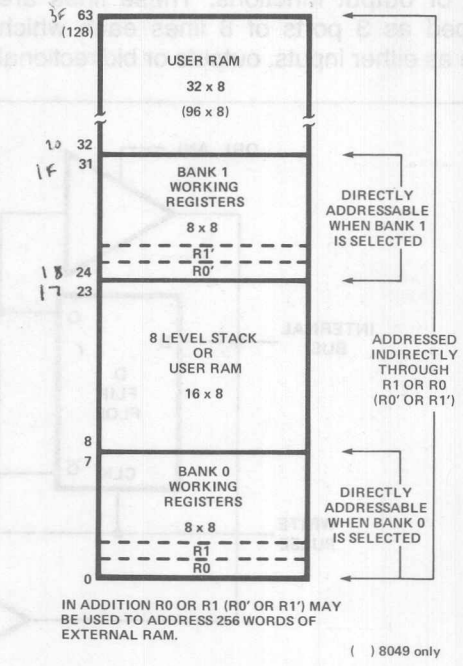


MCS-48™ PROGRAM MEMORY MAP

Data Memory

Resident data memory is organized as 64 or 128 words 8-bits wide. All locations are indirectly addressable through either of two RAM Pointer Registers which reside at address 0 and 1 of the register array. In addition, the first 8 locations (0-7) of the array are designated as working registers and are directly addressable by several instructions. Since these registers are more easily addressed, they are usually used to store frequently accessed intermediate results. The DJNZ instruction makes very efficient use of the working registers as program loop counters by allowing the programmer to decrement and test the register in a single instruction.

By executing a Register Bank Switch instruction (SEL RB) RAM locations 24-31 are designated as the working registers in place of locations 0-7 and are then directly addressable. This second bank of working registers may be used as an extension of the first bank or reserved for use during interrupt service



DATA MEMORY MAP

subroutines allowing the registers of Bank 0 used in the main program to be instantly "saved" by a Bank Switch. Note that if this second bank is not used, locations 24-31 are still addressable as general purpose RAM. Since the two RAM pointer Registers R0 and R1 are a part of the working register array, bank switching effectively creates two more pointer registers (R0' and R1') which can be used with R0 and R1 to easily access up to four separate working areas in Ram at one time. RAM locations (8-23) also serve a dual role in that they contain the program counter stack as explained in Sec. 2.1.6. These locations are addressed by the Stack Pointer during subroutine calls as well as by RAM Pointer Registers R0 and R1. If the level of subroutine nesting is less than 8, all stack registers are not required and can be used as general purpose RAM locations. Each level of subroutine nesting not used provides the user with two additional RAM locations.

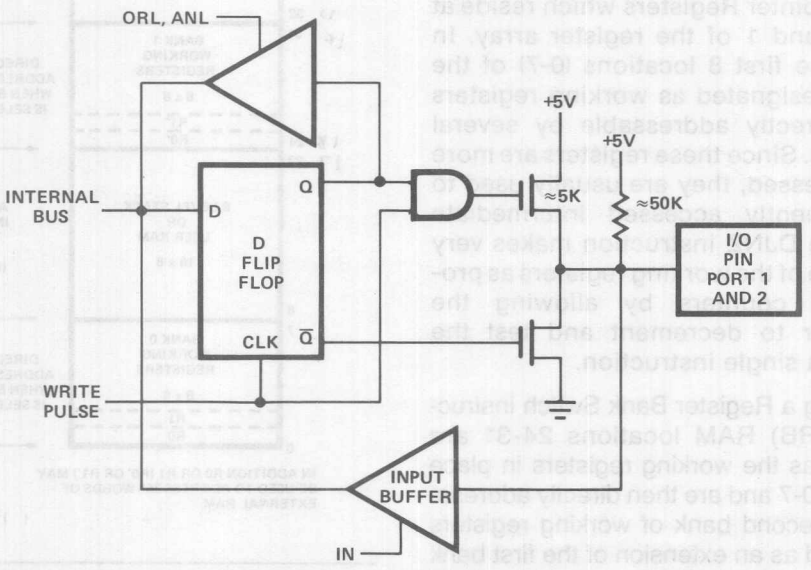
2.1.4 Input/Output

The 8048 has 27 lines which can be used for input or output functions. These lines are grouped as 3 ports of 8 lines each which serve as either inputs, outputs or bidirectional ports and 3 "test" inputs which can alter program sequences when tested by conditional jump instructions.

Ports 1 and 2

Ports 1 and 2 are each 8 bits wide and have identical characteristics. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non latching, i.e., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and outputs will drive one standard TTL load.

The lines of ports 1 and 2 are called quasi-bidirectional because of a special output circuit structure which allows each line to serve as an input, an output, or both even though outputs are statically latched. The figure shows the circuit configuration in detail. Each line is continuously pulled up to +5v through a resistive device of relatively high impedance (~50KΩ). This pullup is sufficient to provide the source current for a TTL high level yet can be pulled low by a standard TTL gate thus allowing the same pin to be used for both input and output. To provide fast switching times in a "0" to "1" transition a relatively low



"QUASI BI DIRECTIONAL" PORT STRUCTURE

impedance device (~5KΩ) is switched in momentarily (~500ns) whenever a "1" is written to the line. When a "0" is written to the line a low impedance (~300Ω) device overcomes the light pullup and provides TTL current sinking capability. Since the pulldown transistor is a low impedance device a "1" must first be written to any line which is to be used as an input. Reset initializes all lines to the high impedance "1" state. This structure allows input and output on the same pin and also allows a mix of input lines and output lines on the same port. The quasi-bidirectional port in combination with the ANL and ORL logical instructions provide an efficient means for handling single line inputs and outputs within an 8-bit processor. See also Section 3.7.

Bus

Bus is also an 8-bit port which is a true bidirectional port with associated input and output strobes. If the bidirectional feature is not needed, Bus can serve as either a statically latched output port or non-latching input port. Input and output lines on this port cannot be mixed however.

As a static port, data is written and latched using the OUTL instruction and inputted using the INS instruction. The INS and OUTL instructions generate pulses on the corresponding $\overline{RD}$ and $\overline{WR}$ output strobe lines; however, in the static port mode they are generally not used. As a bidirectional port the MOVX instructions are used to read and write the port. A write to the port generates a pulse on the $\overline{WR}$ output line and output data is valid at the trailing edge of $\overline{WR}$. A read of the port generates a pulse on the $\overline{RD}$ output line and input data must be valid at the trailing edge of $\overline{RD}$. When not being written or read, the BUS lines are in a high impedance state. See also Sections 3.6 and 3.7.

2.1.5 Test and INT Inputs

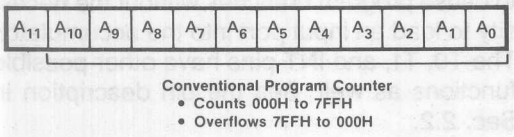
Three pins serve as inputs and are testable with the conditional jump instruction. These are T0, T1, and $\overline{INT}$. These pins allow inputs to cause program branches without the necessity to load an input port into the accumulator. The T0, T1, and $\overline{INT}$ pins have other possible functions as well. See the pin description in Sec. 2.2.

2.1.6 Program Counter and Stack

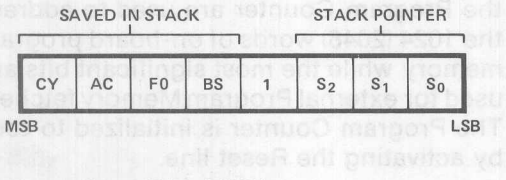
The Program Counter is an independent counter while the Program Counter Stack is implemented using pairs of registers in the Data Memory Array. Only 10 (or 11) bits of the Program Counter are used to address the 1024 (2048) words of on-board program memory while the most significant bits are used for external Program Memory fetches. The Program Counter is initialized to zero by activating the Reset line.

An interrupt or CALL to a subroutine causes the contents of the program counter to be stored in one of the 8 register pairs of the Program Counter Stack. The pair to be used is determined by a 3-bit Stack Pointer which is part of the Program Status Word (PSW). Data RAM locations 8 thru 23 are available as stack registers and are used to store the Program Counter and 4 bits of PSW as shown in the figure. The Stack Pointer when initialized to 000 points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9 of the RAM array. The stack pointer is then incremented by one to point to locations 10 and 11 in anticipation of another CALL. Nesting of subroutines within subroutines can continue up to 8 times without overflowing the stack. If overflow does occur the deepest address stored (location 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111.

The end of a subroutine, which is signalled by a return instruction (RET or RETR), causes the Stack Pointer to be decremented and the contents of the resulting register pair to be transferred to the Program Counter.

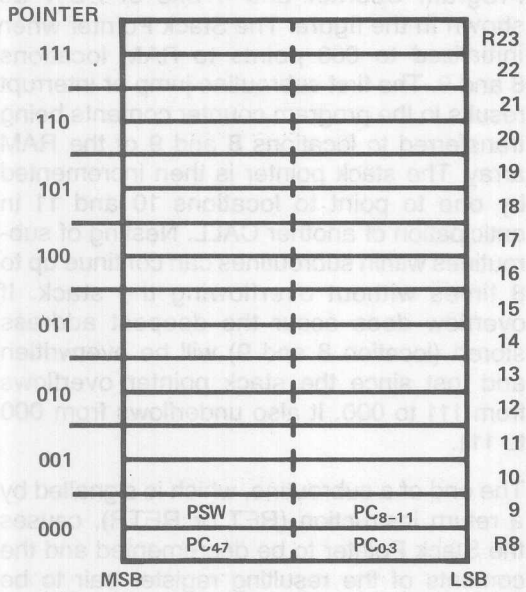


PROGRAM COUNTER



CY CARRY
AC AUXILLARY CARRY
F0 FLAG 0
BS REGISTER BANK SELECT

PROGRAM STATUS WORD (PSW)



PROGRAM COUNTER STACK

2.1.7 Program Status Word

An 8-bit status word which can be loaded to and from the accumulator exists called the Program Status Word (PSW). The accompanying figure shows the information available in the word. The Program Status Word is actually a collection of flip-flops throughout the machine which can be read or written as a whole. The ability to write to PSW allows for easy restoration of machine status after a power down sequence.

The upper four bits of PSW are stored in the Program Counter Stack with every call to subroutine or interrupt vector and are optionally restored upon return with the RETR instruction. The RET return instruction does not update PSW.

The PSW bit definitions are as follows:

Bits 0 - 2: Stack Pointer bits (S_0, S_1, S_2)

Bit 3: Not used ("1" level when read)

Bit 4: Working Register Bank Switch Bit (BS)
0 = Bank 0
1 = Bank 1

Bit 5: Flag 0 bit (F0) user controlled flag which can be complemented or cleared, and tested with the conditional jump instruction JF0.

Bit 6: Auxiliary Carry (AC) carry bit generated by an ADD instruction and used by the decimal adjust instruction DA A.

Bit 7: Carry (CY) carry flag which indicates that the previous operation has resulted in overflow of the accumulator.

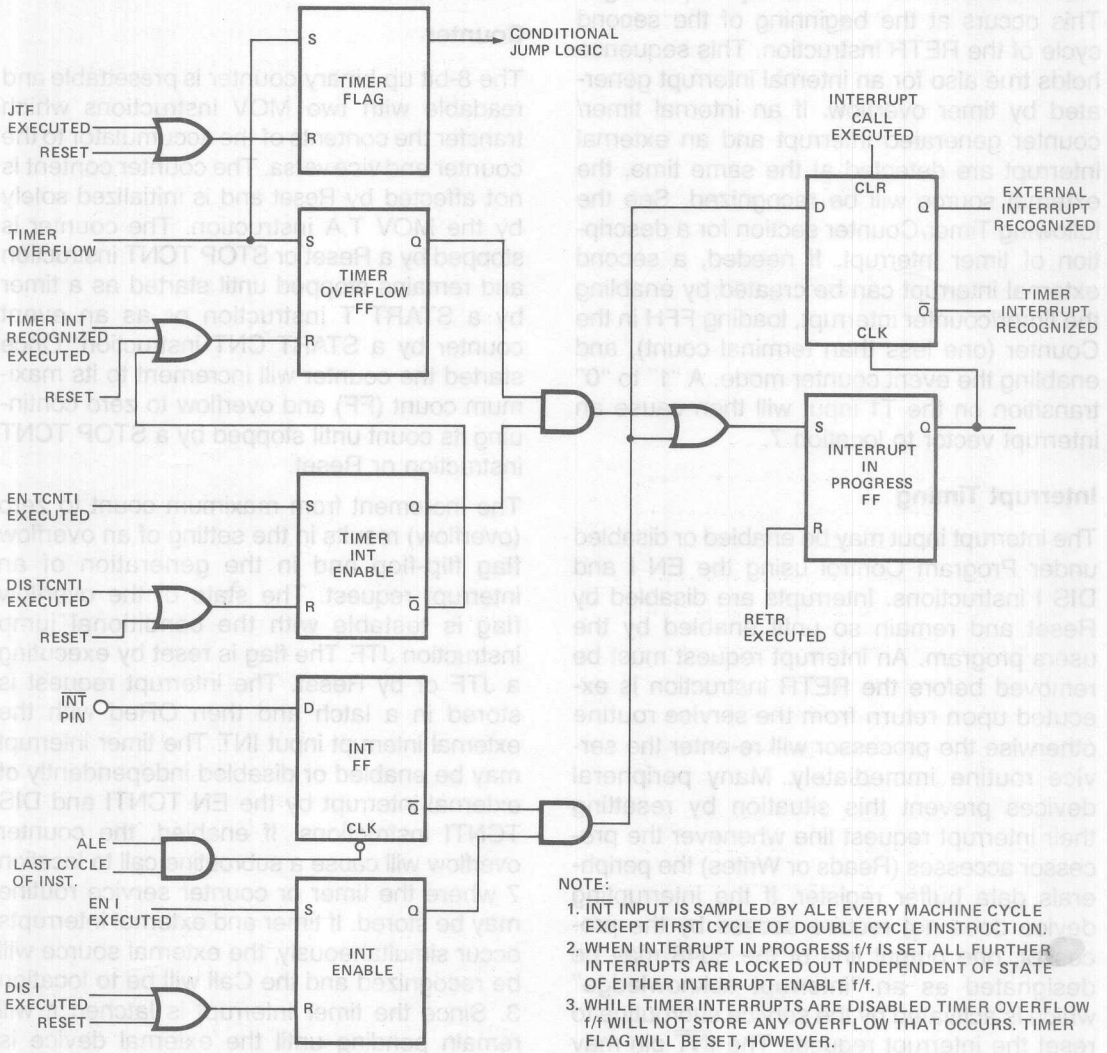
2.1.8 Conditional Branch Logic

The conditional branch logic within the processor enables several conditions internal and external to the processor to be tested by the users program. By using the conditional jump instruction the following conditions can effect a change in the sequence of the program execution.

Device Testable	Jump Conditions (Jump On)	
	All zeros	not all zeros
Accumulator	—	1
Accumulator Bit	0	1
Carry Flag	—	1
User Flags (F0, F1)	—	1
Timer Overflow Flag	0	1
Test Inputs (T0, T1)	0	1
Interrupt Input ($\overline{INT}$)	0	—

2.1.9 Interrupt

An interrupt sequence is initiated by applying a low "0" level input to the $\overline{INT}$ pin. Interrupt is level triggered and active low to allow "WIRE ORing" of several interrupt sources at the input pin. The Interrupt line is sampled every machine cycle during ALE and when detected causes a "jump to subroutine" at location 3 in program memory as soon as all cycles of the current instruction are complete. As in any CALL to subroutine, the Program Counter



NOTE:

1. $\overline{INT}$ INPUT IS SAMPLED BY ALE EVERY MACHINE CYCLE EXCEPT FIRST CYCLE OF DOUBLE CYCLE INSTRUCTION.
2. WHEN INTERRUPT IN PROGRESS f/f IS SET ALL FURTHER INTERRUPTS ARE LOCKED OUT INDEPENDENT OF STATE OF EITHER INTERRUPT ENABLE f/f.
3. WHILE TIMER INTERRUPTS ARE DISABLED TIMER OVERFLOW f/f WILL NOT STORE ANY OVERFLOW THAT OCCURS. TIMER FLAG WILL BE SET, HOWEVER.

INTERRUPT LOGIC

and Program Status word are saved in the stack. For a description of this operation see the previous section, Program Counter and Stack. Program Memory location 3 usually contains an unconditional jump to an interrupt service subroutine elsewhere in program memory. The end of an interrupt service subroutine is signalled by the execution of a Return and Restore Status instruction RETR. The interrupt system is single level in that once an interrupt is detected all further interrupt requests are ignored until execution of an RETR re-enables the interrupt input logic. This occurs at the beginning of the second cycle of the RETR instruction. This sequence holds true also for an internal interrupt generated by timer overflow. If an internal timer/counter generated interrupt and an external interrupt are detected at the same time, the external source will be recognized. See the following Timer/Counter section for a description of timer interrupt. If needed, a second external interrupt can be created by enabling the timer/counter interrupt, loading FFH in the Counter (one less than terminal count), and enabling the event counter mode. A "1" to "0" transition on the T1 input will then cause an interrupt vector to location 7.

Interrupt Timing

The interrupt input may be enabled or disabled under Program Control using the EN I and DIS I instructions. Interrupts are disabled by Reset and remain so until enabled by the users program. An interrupt request must be removed before the RETR instruction is executed upon return from the service routine otherwise the processor will re-enter the service routine immediately. Many peripheral devices prevent this situation by resetting their interrupt request line whenever the processor accesses (Reads or Writes) the peripherals data buffer register. If the interrupting device does not require access by the processor, one output line of the 8048 may be designated as an "interrupt acknowledge" which is activated by the service subroutine to reset the interrupt request. The $\overline{INT}$ pin may also be tested using the conditional jump instruction JNI. This instruction may be used to detect the presence of a pending interrupt before interrupts are enabled. If interrupt is left disabled, $\overline{INT}$ may be used as another test input like T0 and T1.

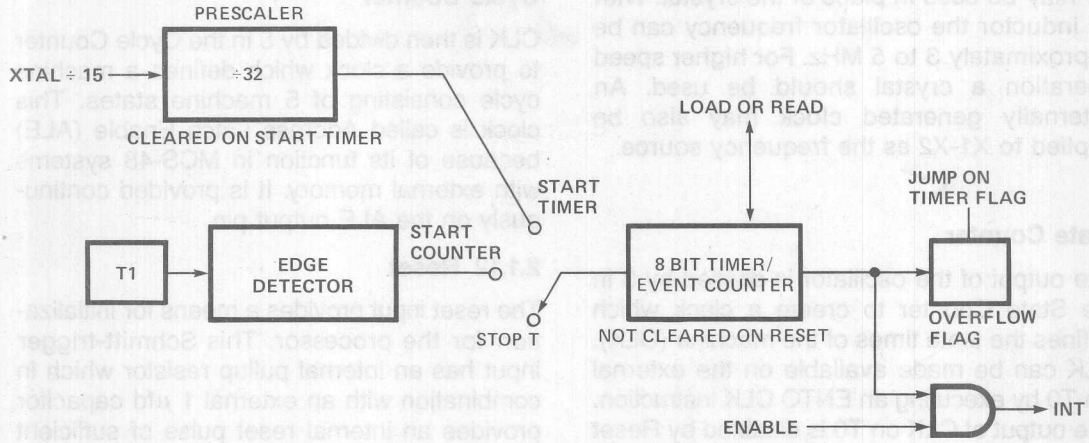
2.1.10 Timer/Counter

The 8048 contains a counter to aid the user in counting external events and generating accurate time delays without placing a burden on the processor for these functions. In both modes the counter operation is the same, the only difference being the source of the input to the counter.

Counter

The 8-bit up binary counter is presettable and readable with two MOV instructions which transfer the contents of the accumulator to the counter and vice versa. The counter content is not affected by Reset and is initialized solely by the MOV T,A instruction. The counter is stopped by a Reset or STOP TCNT instruction and remains stopped until started as a timer by a START T instruction or as an event counter by a START CNT instruction. Once started the counter will increment to its maximum count (FF) and overflow to zero continuing its count until stopped by a STOP TCNT instruction or Reset.

The increment from maximum count to zero (overflow) results in the setting of an overflow flag flip-flop and in the generation of an interrupt request. The state of the overflow flag is testable with the conditional jump instruction JTF. The flag is reset by executing a JTF or by Reset. The interrupt request is stored in a latch and then ORed with the external interrupt input INT. The timer interrupt may be enabled or disabled independently of external interrupt by the EN TCNTI and DIS TCNTI instructions. If enabled, the counter overflow will cause a subroutine call to location 7 where the timer or counter service routine may be stored. If timer and external interrupts occur simultaneously, the external source will be recognized and the Call will be to location 3. Since the timer interrupt is latched it will remain pending until the external device is serviced and immediately be recognized upon return from the service routine. The



TIMER/EVENT COUNTER

pending timer interrupt is reset by the Call to location 7 or may be removed by executing a DIS TCNTI instruction.

As an Event Counter

Execution of a START CNT instruction connects the T1 input pin to the counter input and enables the counter. Subsequent high to low transitions on T1 will cause the counter to increment. The maximum rate at which the counter may be incremented is once per three instruction cycles (every 7.5μsec when using a 6MHz crystal)—there is no minimum frequency. T1 input must remain high for at least 500ns (at 6MHz) after each transition.

As a Timer

Execution of a START T instruction connects an internal clock to the counter input and enables the counter. The internal clock is derived by passing the basic 400 KHz machine cycle clock ALE through a ÷ 32 prescaler. The prescaler is reset during the START T instruction. The resulting 12.5 KHz clock increments the counter every 80 μsec (assuming 6 MHz XTAL). Various delays between 80 μsec and 20 msec (256 counts) can be obtained by presetting the counter and detecting overflow. Times longer than 20 msec may be achieved by accumulating multiple overflows in a register under software control. For time resolution less than 80 μsec an external clock can be applied to the T1 input and the counter operated in the event counter mode. ALE divided by 3 or more can serve as this external clock. Very small delays or "fine tuning" of larger delays can be easily accomplished by software delay loops.

2.1.11 Clock and Timing Circuits

Timing generation for the 8048 is completely self-contained with the exception of a frequency reference which can be XTAL, inductor, or external clock source. The Clock and Timing circuitry can be divided into the following functional blocks:

Oscillator

The on-board oscillator is a high gain series resonant circuit with a frequency range of 1 to 6MHz. The X1 external pin is the input to the amplifier stage while X2 is the output. A crystal or inductor connected between X1 and X2 provides the feedback and phase shift required for oscillation. A 5.9904 MHz crystal provides for easy derivation of all standard communications frequencies. If an accurate frequency reference and maximum processor speed are not required, an induc-

tor may be used in place of the crystal. With an inductor the oscillator frequency can be approximately 3 to 5 MHz. For higher speed operation a crystal should be used. An externally generated clock may also be applied to X1-X2 as the frequency source.

State Counter

The output of the oscillator is divided by 3 in the State Counter to create a clock which defines the state times of the machine (CLK). CLK can be made available on the external pin T0 by executing an ENTO CLK instruction. The output of CLK on T0 is disabled by Reset of the processor.

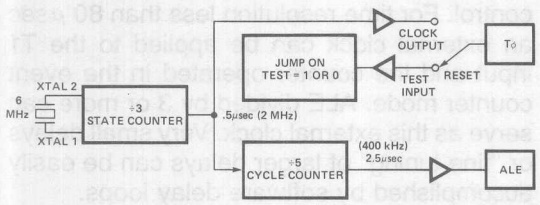
Cycle Counter

CLK is then divided by 5 in the Cycle Counter to provide a clock which defines a machine cycle consisting of 5 machine states. This clock is called Address Latch Enable (ALE) because of its function in MCS-48 systems with external memory. It is provided continuously on the ALE output pin.

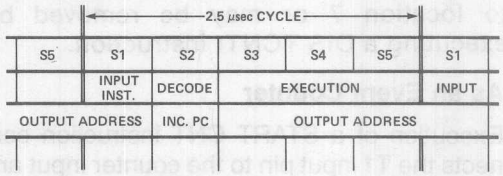
2.1.12 Reset

The reset input provides a means for initialization for the processor. This Schmitt-trigger input has an internal pullup resistor which in combination with an external 1 μfd capacitor provides an internal reset pulse of sufficient length to guarantee all circuitry is reset. If the

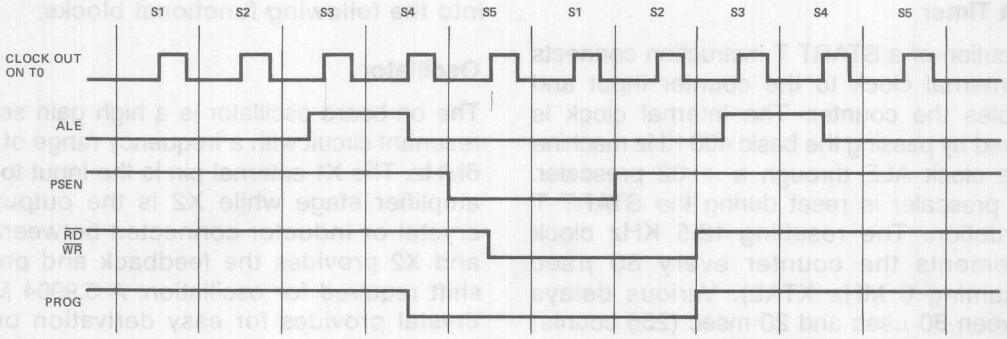
DIAGRAM OF 8048 CLOCK UTILITIES



INSTRUCTION CYCLE



MCS-48™ CYCLE TIMING



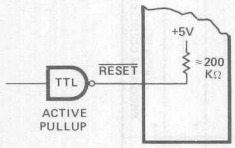
	CYCLE 1					CYCLE 2				
INSTRUCTION	S1	S2	S3	S4	S5	S1	S2	S3	S4	S5
IN A,P	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	——	——	READ PORT	* ——	——	——
OUTL P,A	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	OUTPUT TO PORT	——	——	* ——	——	——
ANL P, =DATA	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	——	* INCREMENT PROGRAM COUNTER	OUTPUT TO PORT	——
ORL P, =DATA	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	——	* INCREMENT PROGRAM COUNTER	OUTPUT TO PORT	——
INS A, BUS	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	——	——	READ PORT	* ——	——	——
OUTL BUS, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	OUTPUT TO PORT	——	——	* ——	——	——
ANL BUS, =DATA	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	——	* INCREMENT PROGRAM COUNTER	OUTPUT TO PORT	——
ORL BUS, =DATA	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	INCREMENT TIMER	READ PORT	FETCH IMMEDIATE DATA	——	* INCREMENT PROGRAM COUNTER	OUTPUT TO PORT	——
MOVX @ R,A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT RAM ADDRESS	INCREMENT TIMER	OUTPUT DATA TO RAM	——	——	* ——	——	——
MOVX A,@R	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT RAM ADDRESS	INCREMENT TIMER	——	——	READ DATA	* ——	——	——
MOVD A, P_i	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	——	——	READ P2 LOWER	* ——	——	——
MOVD P_i, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA TO P2 LOWER	——	——	* ——	——	——
ANLD P, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA	——	——	* ——	——	——
ORLD P, A	FETCH INSTRUCTION	INCREMENT PROGRAM COUNTER	OUTPUT OPCODE/ADDRESS	INCREMENT TIMER	OUTPUT DATA	——	——	* ——	——	——
J (CONDITIONAL)	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	SAMPLE CONDITION	INCREMENT TIMER	——	FETCH IMMEDIATE DATA	——	* UPDATE PROGRAM COUNTER	——	——
STRT T STRT CNT	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	——	START COUNTER					
STOP TCNT	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	——	STOP COUNTER					
ENI	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	ENABLE INTERRUPT	——					
DIS I	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	DISABLE INTERRUPT	——					
ENTO CLK	FETCH INSTRUCTION	* INCREMENT PROGRAM COUNTER	——	ENABLE CLOCK	——					

*VALID INSTRUCTION ADDRESSES ARE OUTPUT AT THIS TIME IF EXTERNAL PROGRAM MEMORY IS BEING ACCESSED.

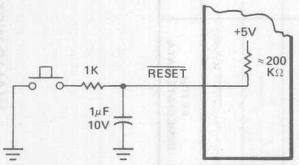
ICTION TIMING DIAGRAM

reset pulse is generated externally the reset pin must be held at ground (.5V) for at least 50 milliseconds after the power supply is within tolerance. Only 5 machine cycles (12.5μs @ 6 MHz) are required if power is already on and the oscillator has stabilized.

EXTERNAL RESET



POWER ON RESET



Reset performs the following functions:

1. Sets program counter to zero.
2. Sets stack pointer to zero.
3. Selects register bank 0.
4. Selects memory bank 0.
5. Sets BUS to high impedance state. (except when EA = 5V)
6. Sets Ports 1 and 2 to input mode.
7. Disables interrupts (timer and external)
8. Stops timer.
9. Clears timer flag.
10. Clears F0 and F1.
11. Disables clock output from T0.

2.1.13 Single-Step

This feature provides the user with a debug capability in that the processor can be stepped through the program one instruction at a time. While stopped, the address of the next instruction to be fetched is available concurrently on BUS and the lower half of Port 2. The user can therefore follow the program through each of the instruction steps. A timing diagram, showing the interaction between output ALE and input SS is shown. The BUS buffer contents are lost during single step, however, a latch may be added to re-establish the lost I/O capability if needed. (See 2.4.1).

Timing

The 8048 operates in a single-step mode as follows:

1. The processor is requested to stop by applying a low level on $\overline{SS}$.

2. The processor responds by stopping during the instruction fetch portion of the next instruction. If a double cycle instruction is in progress when the single step command is received, both cycles will be completed before stopping.

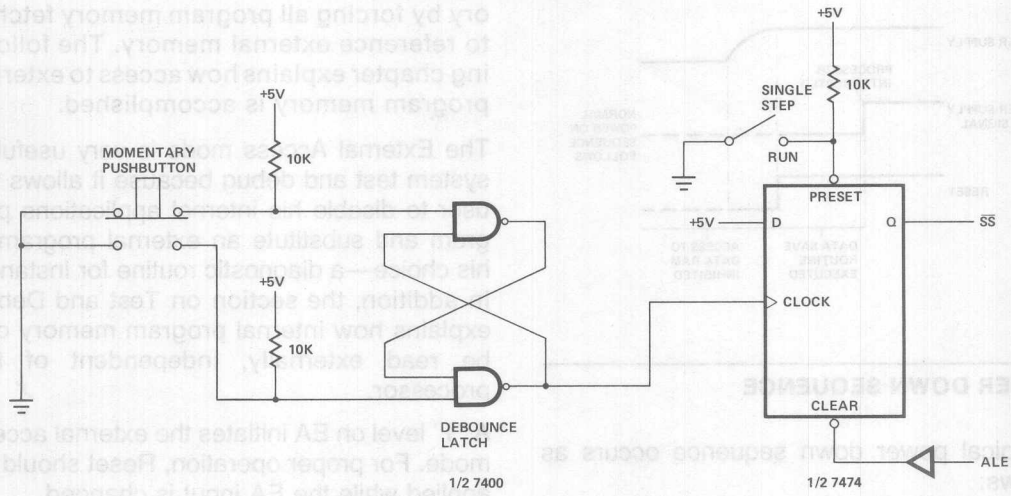
3. The processor acknowledges it has entered the stopped state by raising ALE high. In this state (which can be maintained indefinitely) the address of the next instruction to be fetched is present on BUS and the lower half of port 2.

4. $\overline{SS}$ is then raised high to bring the processor out of the stopped mode allowing it to fetch the next instruction. The exit from stop is indicated by the processor bringing ALE low.

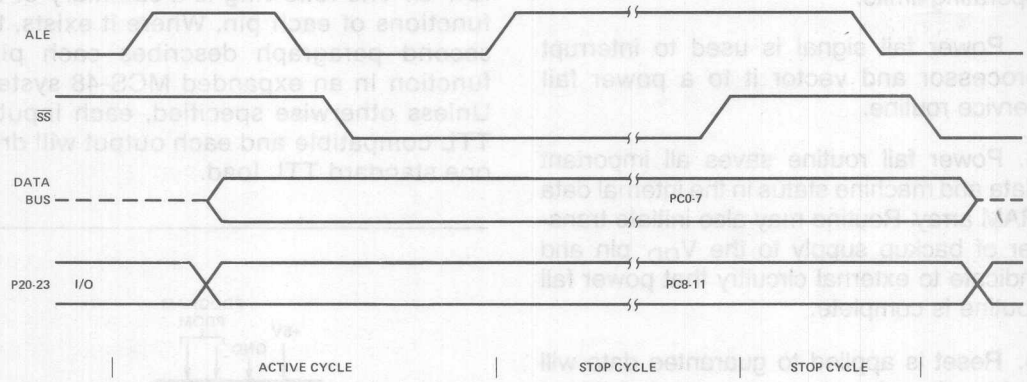
5. To stop the processor at the next instruction $\overline{SS}$ must be brought low again as soon as ALE goes low. If $\overline{SS}$ is left high the processor remains in a "Run" mode.

A diagram for implementing the single step function of the 8748 is shown. A D-type flip-flop with preset and clear is used to generate $\overline{SS}$. In the run mode $\overline{SS}$ is held high by keeping the flip-flop preset (preset has precedence over the clear input). To enter single step, preset is removed allowing ALE to bring $\overline{SS}$ low via the clear input. ALE should be buffered since the clear input of an SN7474 is the equivalent of 3 TTL loads. The processor is now in the stopped state. The next instruction is initiated by clocking a "1" into the flip-flop. This "1" will not appear on $\overline{SS}$ unless ALE is high removing clear from the flip-flop. In response to $\overline{SS}$ going high the processor begins an instruction fetch which brings ALE low resetting $\overline{SS}$ through the clear input and causing the processor to again enter the stopped state.

SINGLE STEP CIRCUIT



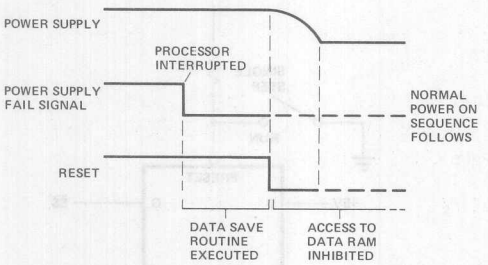
SINGLE STEP TIMING



2.1.14 Power Down Mode (8048, 8049, 8039, 8035L)

Extra circuitry has been added to the 8048 ROM version to allow power to be removed from all but the 64/128 x 8 data RAM array for low power standby operation. In the power down mode the contents of data RAM can be maintained while drawing typically 10% to 15% of normal operating power requirements.

V_{CC} serves as the 5V supply pin for the bulk of 8048 circuitry while the V_{DD} pin supplies only the RAM array. In normal operation both pins are at 5V while in standby V_{CC} is at ground and only V_{DD} is maintained at 5V. Applying Reset to the processor through the Reset pin inhibits any access to the RAM by the processor and guarantees that RAM data will not be inadvertently altered as V_{CC} is removed from V_{CC}.



POWER DOWN SEQUENCE

A typical power down sequence occurs as follows:

1. Imminent power supply failure is detected by user defined circuitry. Signal must be early enough to allow 8048 to save all necessary data before V_{CC} falls below normal operating limits.

2. Power fail signal is used to interrupt processor and vector it to a power fail service routine.

3. Power fail routine saves all important data and machine status in the internal data RAM array. Routine may also initiate transfer of backup supply to the V_{DD} pin and indicate to external circuitry that power fail routine is complete.

4. Reset is applied to guarantee data will not be altered as the power supply falls out of limits. Reset must be held low until V_{CC} is at ground level.

Recovery from the Power Down mode can occur as any other power-on sequence with an external capacitor on the Reset input providing the necessary delay. See the previous section on Reset.

2.1.15 External Access Mode

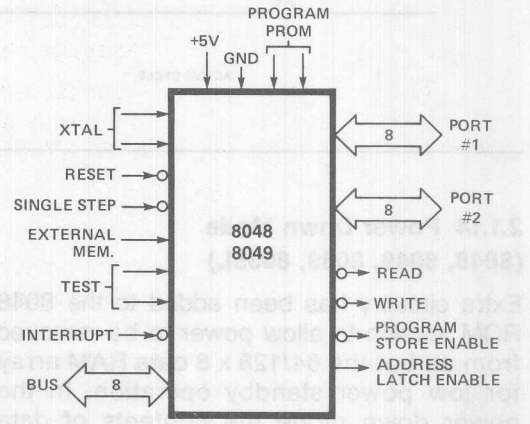
Normally the first 1K (8048) or 2K (8049) words of program memory are automatically fetched from internal ROM or EPROM. The EA input pin however allows the user to effectively disable internal program memory by forcing all program memory fetches to reference external memory. The following chapter explains how access to external program memory is accomplished.

The External Access mode is very useful in system test and debug because it allows the user to disable his internal applications program and substitute an external program of his choice—a diagnostic routine for instance. In addition, the section on Test and Debug explains how internal program memory can be read externally, independent of the processor.

A "1" level on EA initiates the external access mode. For proper operation, Reset should be applied while the EA input is changed.

2.2 Pin Description

The MCS-48 processors (except 8021) are packaged in 40 pin Dual In-Line Packages (DIP's). The following is a summary of the functions of each pin. Where it exists, the second paragraph describes each pin's function in an expanded MCS-48 system. Unless otherwise specified, each input is TTL compatible and each output will drive one standard TTL load.



8048 LOGIC SYMBOL

Designation	Pin Number	Function
V_{SS}	20	Circuit GND potential
V_{DD}	26	Programming power supply; +25V during program, +5V during operation for both ROM and PROM. Low power standby pin in 8048 ROM version
V_{CC}	40	Main power supply; +5V during operation and 8748 programming.
PROG	25	Program pulse (+25V) input pin during 8748 programming.
		Output strobe for 8243 I/O expander.
P10-P17 (Port 1)	27-34	8-bit quasi-bidirectional port. (Internal Pullup ≈ 50KΩ)
P20-P27 (Port 2)	21-24 35-38	8-bit quasi-bidirectional port. (Internal Pullup ≈ 50KΩ)
		P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.
D0-D7 (BUS)	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched.
		Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of PSEN. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENTO CLK instruction. T0 is also used during programming.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the event counter input using the STRT CNT instruction.
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. (Active low)
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the BUS from an external device. (Active low)
		Used as a Read Strobe to External Data Memory.

Designation	Pin Number	Function
$\overline{\text{RESET}}$	4	Input which is used to initialize the processor. Also used during PROM programming and verification. (Active low) (Internal pullup ≈ 200KΩ)
$\overline{\text{WR}}$	10	Output strobe during a BUS write. (Active low) Used as write strobe to external data memory.
ALE	11	Address Latch Enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.
$\overline{\text{PSEN}}$	9	Program Store Enable. This output occurs only during a fetch to external program memory. (Active Low)
$\overline{\text{SS}}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction. (Active Low) (Internal pullup ≈ 300KΩ)
EA	7	External Access input which forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active High) (Internal pullup ≈ 10MΩ on 8048/8049, 8035L, 8039 only)
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source.
XTAL2	3	Other side of crystal input.

Unless otherwise stated inputs do not have internal pullup resistors.

2.3 Programming, Verifying and Erasing EPROM

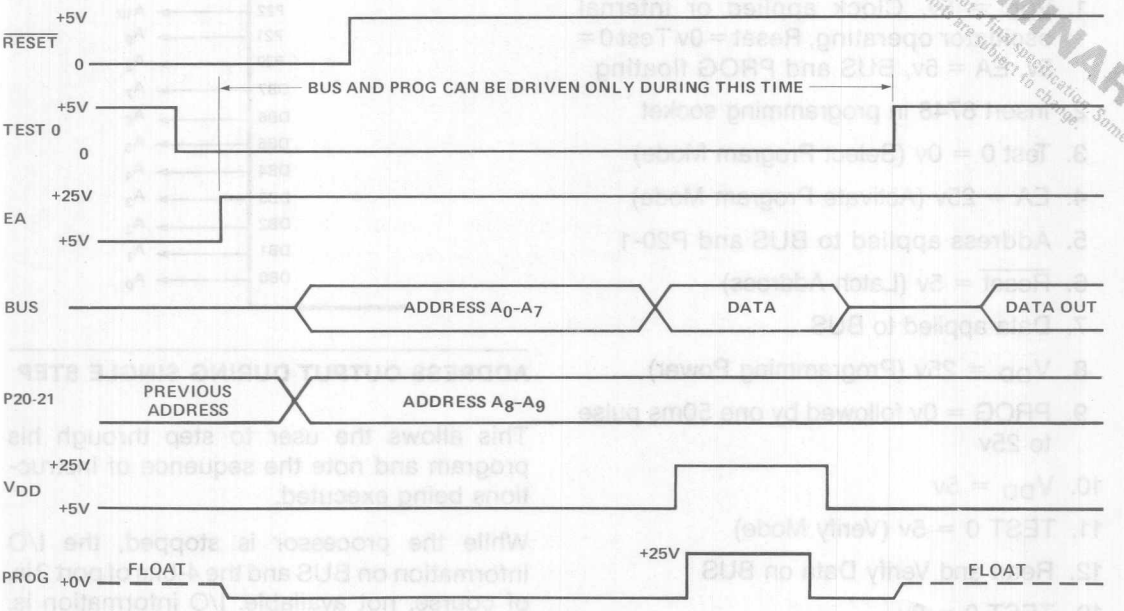
The internal Program Memory of the 8748 may be erased and reprogrammed by the user as explained in the following sections:

2.3.1 Programming/Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input (1 to 6MHz)
$\overline{\text{Reset}}$	Initialization and Address Latching
Test 0	Selection of Program or Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input Data Output During Verify
P20-1	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.



SEE 8048/8748 DATA SHEET (CHAPTER 6) FOR DETAIL TIMING SPECIFICATIONS.

WARNING: An attempt to program a mis-socketed 8748 will result in severe damage to the part. An indication of a properly socketed part is the appearance of the ALE clock output. The lack of this clock may be used to disable the programmer.

PROGRAMMING/VERIFY SEQUENCE

8748 Erasure Characteristics

The erasure characteristics of the 8748 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 8748 in approximately 3 years while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 8748 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8748 window to prevent unintentional erasure.

When erased, bits of the 8748 Program Memory are in the logic "0" state.

The recommended erasure procedure for the 8748 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000μW/cm^2 power rating. The 8748 should be placed within one inch from the lamp tubes during erasure. Some lamps have a filter on their tubes and this filter should be removed before erasure.

The Program/Verify sequence is:

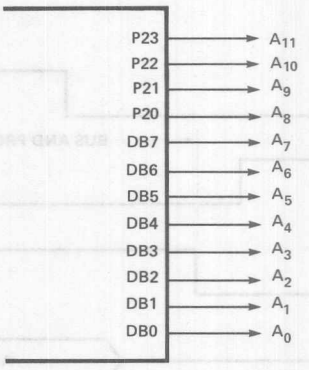
1. V_{DD} = 5v, Clock applied or internal oscillator operating, Reset = 0v Test 0 = 5v, EA = 5v, BUS and PROG floating
2. Insert 8748 in programming socket
3. Test 0 = 0v (Select Program Mode)
4. EA = 25v (Activate Program Mode)
5. Address applied to BUS and P20-1
6. $\overline{\text{Reset}}$ = 5v (Latch Address)
7. Data applied to BUS
8. V_{DD} = 25v (Programming Power)
9. PROG = 0v followed by one 50ms pulse to 25v
10. V_{DD} = 5v
11. TEST 0 = 5v (Verify Mode)
12. Read and Verify Data on BUS
13. TEST 0 = 0v
14. $\overline{\text{Reset}}$ = 0v and repeat from Step 5
15. Programmer should be at conditions of Step 1 when 8748 is removed from socket.

2.4 Test and Debug

Several MCS-48 features described in the previous sections are discussed here to emphasize their use in testing MCS-48 components and in debugging MCS-48 based systems.

2.4.1 Single Step

Single step circuitry within the microcomputer in combination with the external circuitry described in Section 2.1.13 allows the user to execute one instruction at a time whether the instruction is one or two cycles in length. After completion of the instruction the processor halts with the address of the next instruction to be fetched available on the eight lines of BUS and the lower 4-bits of port 2.



ADDRESS OUTPUT DURING SINGLE STEP

This allows the user to step through his program and note the sequence of instructions being executed.

While the processor is stopped, the I/O information on BUS and the 4-bits of port 2 is, of course, not available. I/O information is, however, valid at the leading edge of ALE and can be latched externally using this signal if necessary.

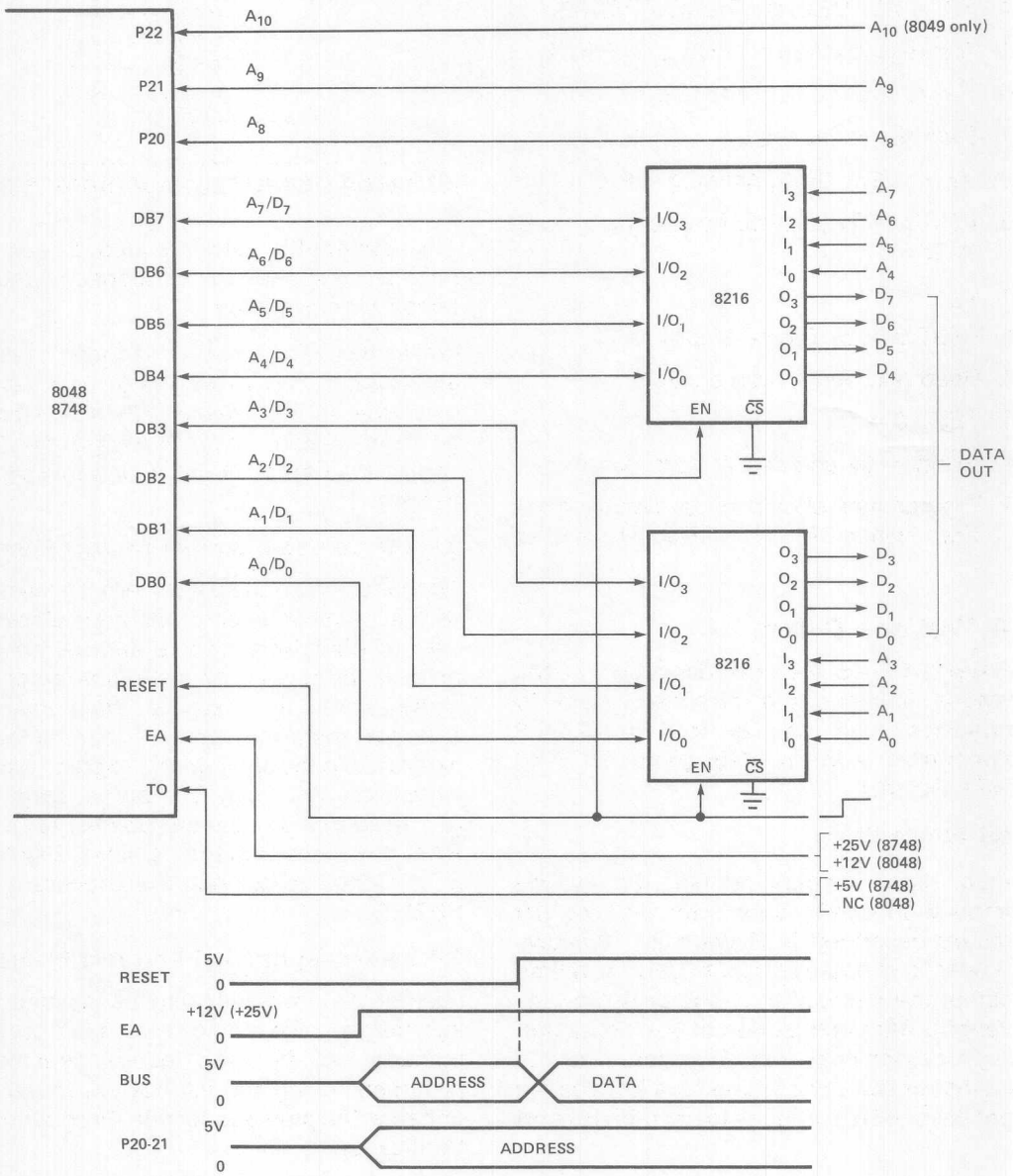
2.4.2 Disabling Internal Program Memory

Applying +5V to the EA (external access) pin of the MCS-48 microcomputers allows the user to effectively disable internal program memory by forcing all instruction fetches to occur from an external memory. This external memory can be connected as explained in the section on program memory expansion and can contain a diagnostic routine to exercise the processor, the internal RAM, the timer, and the I/O lines. EA should be switched only when the processor is in RESET.

2.4.3 Reading Internal Program Memory

Just as the processor may be isolated from internal program memory using EA, program memory can be read independent of the processor using the verification mode described in the previous section, Programming/Verification.

The processor is placed in the READ mode by applying a high voltage (+25V for the 8748, +12V for the 8048/8049) to the EA pin and +5V to the T0 (8748 only) input pin. RESET must be at 0V when voltage is applied to EA. The address of the location to be read is then applied to the same lines (TTL levels) of BUS and Port 2 which output the address during single step (see below). The address is latched by a "0" to "1" transition on RESET and a high level on RESET causes the contents of the program memory location addressed to appear on the eight lines of BUS.



READING INTERNAL PROGRAM MEMORY

8021 Functional Specifications

The following is a functional description of the major elements of the 8021.

2.5 Program Memory

The 8021 contains 1K x 8 of mask programmable ROM. No external ROM expansion capability is provided.

2.6 Data Memory

A 64 x 8 dynamic RAM is located on chip for data storage. All locations are indirectly addressable and eight designated locations are directly addressable. Also, included in the memory is the address stack, addressed by a 3-bit stack pointer.

Memory is organized as shown in Figure 1. The least significant 8 addresses, 0-7, are directly addressable by any of the 11 direct register instructions. The locations are readily accessible for a variety of operations with the least number of instruction bytes required for their manipulation.

Registers 0 and 1 have another function, in that they can be used to indirectly address all locations in memory, using the indirect register instructions. These indirect RAM address registers, IRAR's, are especially useful for repetitive-type operations on adjacent memory locations. The indirect register instruction specifies which IRAR to use, and the contents of the IRAR is used to address a location in RAM. The contents of the addressed location is used during the execution of the instruction and may be modified. A value larger than 63 should not be preset in the IRAR when selected by an indirect register instruction. IRAR's may point to address 0-7, if desired.

Locations 8-23 may be used as the address stack. The address stack enables the processor to keep track of the return addresses generated from CALL instructions. A 3-bit stack pointer (SP) supplies the address of the locations to be loaded with the next return address generated. The SP to this pushdown stack is incremented by one after a return address is stored, and decremented by one before an address is fetched during a RET. A total of 8 levels of nesting is possible. The SP is initialized to location 8 upon RESET. Since each address is 10-bits long, two bytes must be used to store a single address. The SP is incremented and decremented by one, but each increment or decrement moves the address pointed to by two. Therefore, only even numbered addresses are pointed to.

If a particular application does not require 8 levels of nesting, the unused portion of the stack may be used as any other indirectly addressable scratchpad location. For example, if only 3 levels of subroutine nesting are used, then only locations 8-13 need be reserved for the address stack, and locations 14-63 can be used for data storage. The unincremented program counter address is stored in the address stack. The stack contents is incremented before being loaded into the program counter during a return from subroutine.

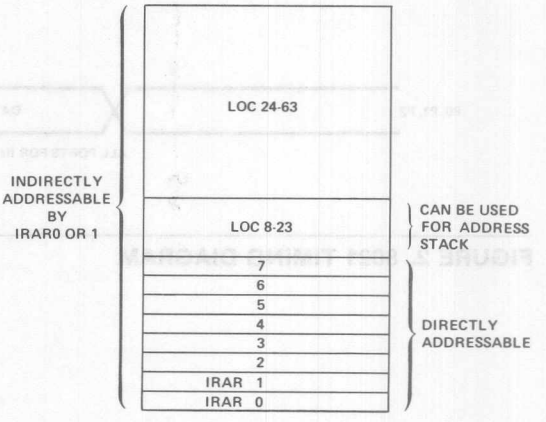


FIGURE 1. INTERNAL RAM ORGANIZATION

2.7 Oscillator and Clock

The 8021 contains its own onboard oscillator and clock circuit, requiring only an external timing control element. This control element can be a crystal, inductor, resistor, or clock in. The capacitor normally required in resistor or inductor timing control operation is integrated onto the 8021. All internal time slots are derived from the external element, and all outputs are a function of the oscillator frequency. Pins XTAL1 and XTAL2 are used to input the particular control element. An instruction cycle consists of 10 states, and each state is a time slot of 3 oscillator periods. (See Figure 2) Therefore, to obtain a 10 μsec instruction cycle, a 3 MHz crystal should be used. An oscillator frequency of 3 MHz may also be obtained by connecting a 10KΩ resistor between XTAL1 and XTAL2. Note that the required resistance may vary from 10KΩ, and should be adjusted as necessary.

The 8021 utilizes dynamic RAM and certain other dynamic logic. Due to the clocking required with dynamic circuits, the oscillator frequency must be equal to or greater than 600K Hz, or improper operation may occur.

2.8 Timer/Event Counter

The 8021 has internal timer/event counter circuits that can monitor elapsed time or count external events that occur during program execution. The circuit has an 8-bit binary up-counter that is presettable and readable with two MOV instructions. These instructions transfer the contents of the accumulator to the counter and vice-versa. The counter content is not affected by Reset, and is initialized solely by the MOV T,A instruction. The counter is stopped by a RESET or STOP TCNT instruction and remains stopped until started as a timer by a STRT T instruction or as an event counter by a STRT CNT instruction. Once started,

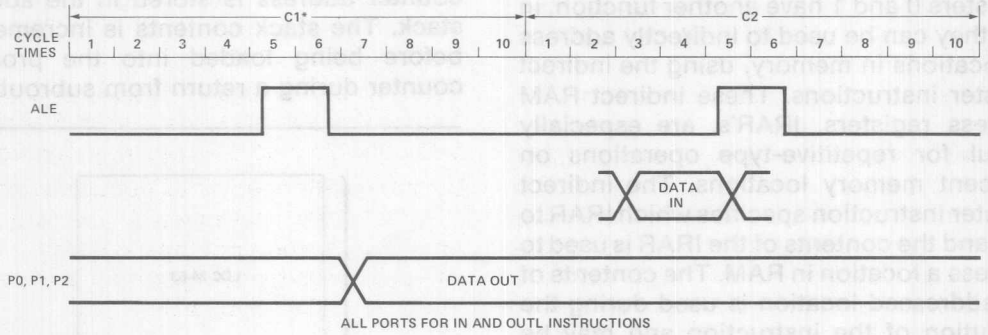


FIGURE 2. 8021 TIMING DIAGRAM

the counter increments to its maximum count (FF), and overflows to zero. The count continues until stopped by a STOP TCNT instruction or RESET. The increment from maximum count to zero (overflow) sets an overflow flag. The state of the overflow flag is testable with the conditional jump instruction JTF. The flag is reset by JTF or by executing a RESET.

By a MOV T,A instruction, the contents of the accumulator are loaded to the timer. At the STRT T command an internal prescaler is zeroed and thereafter increments once each 30 input clocks (once each single cycle instruction, twice each double cycle instruction). The prescaler is a divide by 32. At the (11111) to (00000) transition the timer is incremented. The timer is 8-bits and an overflow (FFH) to (00H) timer flag is set. A conditional branch instruction (JTF) is available for testing this flag, the flag being reset each test. Total count capacity for the timer is $2^8 \times 2^5 = 8192$ or 81.9 msec at a 10 μsec cycle time. Contents of the timer are moved to the accumulator by the MOV A,T instruction without disturbing the counting process. The timer stops upon the STOP TCNT instruction.

The STRT CNT instruction connects the T1 input pin to the event counter input and enables the counter. Subsequent high-to-low transitions on T1 increment the counter. The maximum rate at which the counter can increment is once per three instruction cycles (30μs for a 3 MHz oscillator). There is no minimum frequency. T1 input must remain high for at least 500ns after each transition. The event counter is stopped by a STOP TCNT instruction.

2.9 Input/Output Capabilities

The 8021 I/O configurations are highly flexible. A number of different configurations are possible, tailoring an 8021 to a given task. Other than the power supply and dedicated pins, all other pins (20) can be used for input, output, or both, depending on the configuration.

All ports are quasi-bidirectional to facilitate stand-alone use. A simplified schematic of the quasi-bidirectional interface is shown in Figure 3. This configuration allows buffered outputs, and also allows external input. Data written to these ports is statically latched and remains unchanged until rewritten. As input ports these lines are non-latching, i.e., inputs must be present until read by an input instruction. When writing a "0" or low value to these ports, the large pulldown device sinks an external TTL load. When writing a "1", a large current is supplied through the large pullup device to allow a fast data transfer. After a short time (less than one instruction cycle), the large device is shut off and the small pullup maintains the "1" level indefinitely. However, in this situation, an input device capable of overriding the small amount of sustaining current supplied by the pullup device can be read. (Alternatively, the data written can be read). So, by writing a "1" to any particular pin, that pin can serve either as a true high-level latched output pin, or as just a pullup resistor on an input. This allows maximum user flexibility in selecting his input or latched output pins, with a minimum of external components.

Port 00-07 is also quasi-bidirectional, except there is no large pullup device. As outputs, this port is essentially open drain.

By mask option the small pullup devices on P00-P07 may be deleted on any pin providing a true open drain output. This is useful in driving analog circuits and certain loads, such as keyboards.

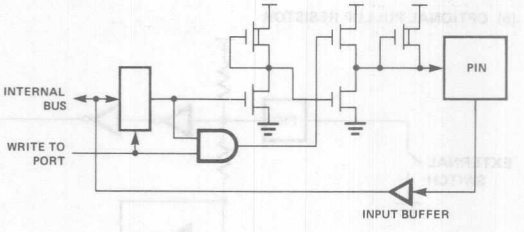


FIGURE 3. QUASI-BIDIRECTIONAL PORT STRUCTURE

2.9.1 T1 Input

The 8021 T1 input line can be used as an input for the following functions:

- Event Counter (external input)
- Test input for branch instructions
- Zero voltage crossing detection

The operation of T1 as an input to the Event Counter is described in the Timer/Event Counter section. When used as a test input, the JT1 and JNT1 instructions test for 1 and 0 levels, respectively.

The T1 pin can also be used to detect the zero crossing of slowly moving AC signals (60 Hz). The self-biasing circuit shown in Figure 4 permits the Test 1 input to detect when the input voltage crosses zero within ±5%, then the voltage is coupled through a 1.0μf capacitor. Maximum input voltage is 3V peak-to-peak. The zero cross detection is especially useful in SCR control of 60 Hz power and in developing time-of-day and other timing routines. As a ROM mask option there is a pullup resistor that is useful for switch contact input or standard TTL.

2.9.2 High Current Outputs

Very high current drive is desirable for minimizing external parts required to do high power control. P10 and P11 have been designated high drive outputs capable of sinking 7mA at V_{SS} + 2.5 volts. (For clarity, this is 7mA to V_{SS} with a 2.5 volt drop across the buffer.) These pins may, of course, be paralleled for 14mA drive if the output logic states are always the same.

2.9.3 Expanded I/O

The 8021 can be used with the 8243 I/O expander chip, which provides additional I/O capability with a limited number of overhead pins. This chip has 4 directly addressable 4-bit ports. It connects to the PROG pin, which provides a clock, and pins P20-P23, which provide address and data. These ports can be written with a MOVD P,A; ANLD P,A; and ORLD P,A for Ports 4-7. A high to low transition on PROG signifies that address and control are available on P20-P23. The previous data on P20-P23 before an output expander instruction is lost. Therefore, when using an output expander P20-P23 are not useful for general input/output. Reading is via the MOVD A,P. This circuit configuration is shown in Figure 5. The timing diagram is shown in Figure 6.

The 8021 can also use standard low cost TTL to expand the number of I/O lines. An example is shown in the Applications section.

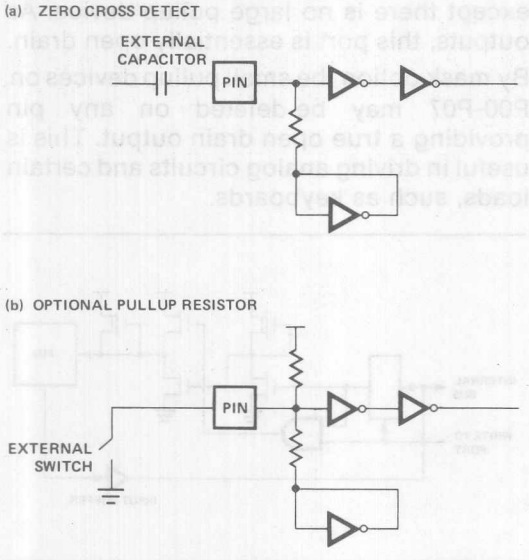


FIGURE 4. TEST 1 PIN

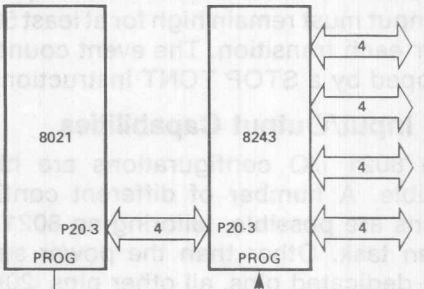


FIGURE 5. I/O EXPANDER INTERFACE

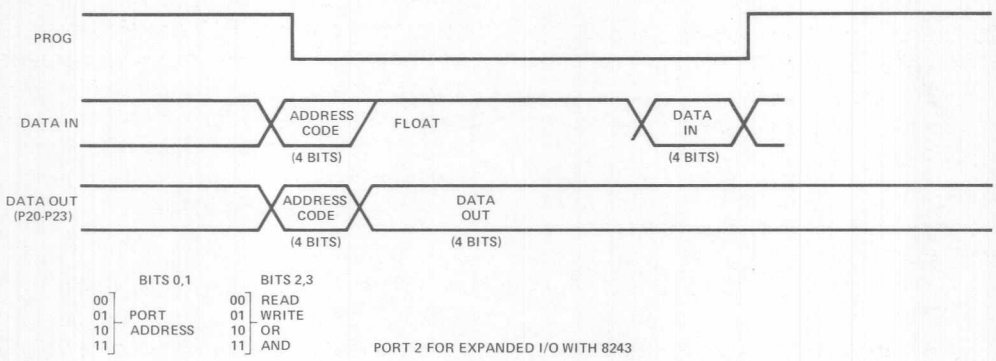


FIGURE 6. EXPANDED I/O TIMING DIAGRAM

2.10 CPU

The 8021 CPU has arithmetic and logical capability. A wide variety of arithmetic and logic instructions may be exercised, which affect the contents of the accumulator, and/or direct or indirect scratchpad locations. Provisions have been made for simplified BCD arithmetic capability using the DAA, SWAP A, and XCHD instructions. In addition, MOVP A,@A allows table lookup for display formating and constants. The conditional branch logic within the processor enables several conditions internal and external to the processor to be tested by the users program. Use the conditional jump instructions with the tests listed below to effect a change in the program execution sequence.

Test	Jump Condition		Jump Instructions
Accumulator	A=0	A≠0	JZ JNZ
Carry Flag	0	1	JNC, JC
Timer Overflow Flag	—	1	JTF
Test Input-T1	0	1	JNT1, JT1

2.11 Reset

A positive-going signal to the RESET input resets the necessary miscellaneous flip-flops and sets the program counter and stack pointer to zero.

The 8021 may see poorly regulated and noisy power supplies. A useful feature is to sense when the power supply dips and do a RESET to prevent continued operation with incorrect data. This feature may be implemented on the 8021 by connecting a diode between the RESET node and ground. See Figure 7.

A RESET will then be forced if the supply drops approximately 1.5 volts and rapidly recovers. One instruction cycle will RESET the 8021 to the initialized state.

By removing the diode and using only the capacitor, voltage drops in V_{CC} will not cause a RESET.

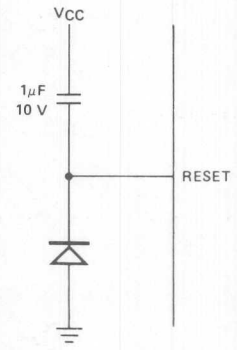


FIGURE 7. POWER ON RESET

Chapter 3

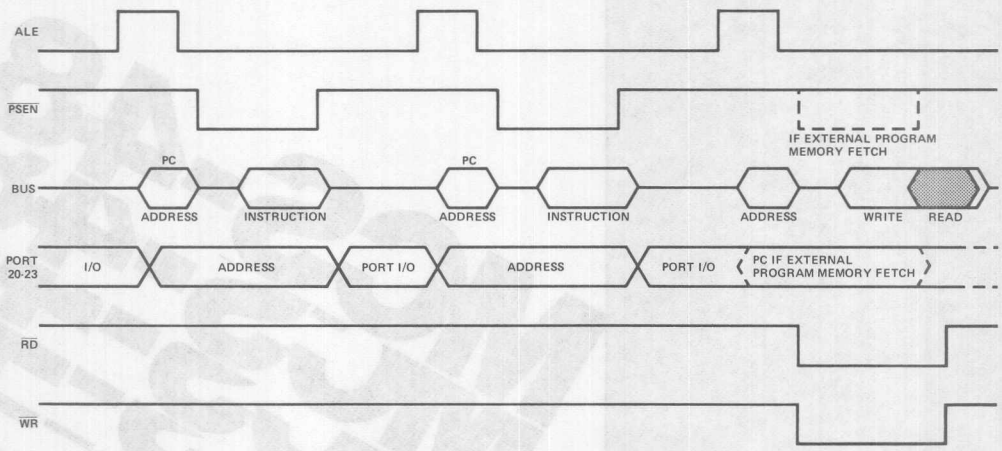
THE EXPANDED MCS-48™ SYSTEM



THE EXPANDED MCS-48™ SYSTEM

3.0 Summary 3-1
3.1 Expansion of Program Memory 3-1
3.2 Expansion of Data Memory 3-4
3.3 Expansion of Input/Output 3-5
3.4 Multi-Chip MCS-48 Systems 3-9
3.5 Memory Bank Switching 3-10
3.6 Control Signal Summary 3-11
3.7 Port Characteristics 3-11

MCS-48™ CYCLE TIMING FOR EXTERNAL MEMORY



THE EXPANDED MCS-48™ SYSTEM

3.0 Summary

If the capabilities resident on the single-chip 8048/8049, 8748, or 8035/8039 are not sufficient for your system requirements, special on-board circuitry allows the addition of a wide variety external memory, I/O, or special peripherals you may require. The processors can be directly and simply expanded in the following areas:

- Program Memory to 4K words
- Data Memory to 320 words (384 words with 8049)
- I/O by unlimited amount
- Special Functions using 8080/8085 peripherals

By using bank switching techniques maximum capability is essentially unlimited. Bank switching is discussed later in the chapter. Expansion is accomplished in two ways:

1. Expander I/O—A special I/O Expander circuit the 8243 provides for the addition of four 4-bit Input/Output ports with the sacrifice of only the lower half (4 bits) of port 2 for inter-device communication. Multiple 8243's may be added to this 4-bit bus by generating the required "chip select" lines.

2. Standard 8085 Bus—One port of the 8048 is like the 8 bit bidirectional data bus of the 8085 microcomputer system allowing interface to the numerous standard memories and peripherals of the MCS-80/85 microcomputer family.

MCS-48 systems can be configured using either or both of these expansion features to optimize system capabilities to the application. Both expander devices and standard memories and peripherals can be added in virtually any number and combination required.

3.1 Expansion of Program Memory

Program Memory is expanded beyond the resident 1K or 2K words by using the 8085 BUS feature of the MCS-48. All program memory fetches from addresses less than 1024 (2048) occur internally with no external signals being generated (except ALE which is always present). At address 1024 the 8048 automatically initiates external program memory fetches.

3.1.1 Instruction Fetch Cycle (External)

For all instruction fetches from addresses of 1024 (2048) or greater the following will occur:

1. The contents of the 12 bit program counter will be output on BUS and the lower half of port 2.

2. Address Latch Enable (ALE) will indicate the time at which address is valid. The trailing edge of ALE is used to latch the address externally.

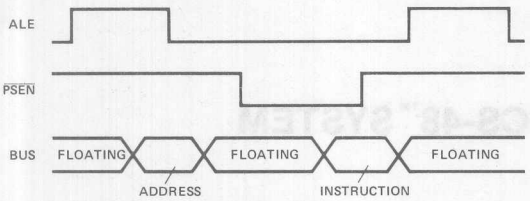
3. Program Store Enable ($\overline{\text{PSEN}}$) indicates that an external instruction fetch is in progress and serves to enable the external memory device.

4. BUS reverts to input (floating) mode and the processor accepts its 8 bit contents as an instruction word.

All instruction fetches including internal addresses can be forced to be external by activating the EA pin of the 8048/8049. The 8035/8039 processors without program memory always operate in the external program memory mode (EA=5V).

3.1.2 Extended Program Memory Addressing (Beyond 2K)

For programs of 2K words or less, the 8048/8049 addresses program memory in



INSTRUCTION FETCH FROM EXTERNAL PROGRAM MEMORY

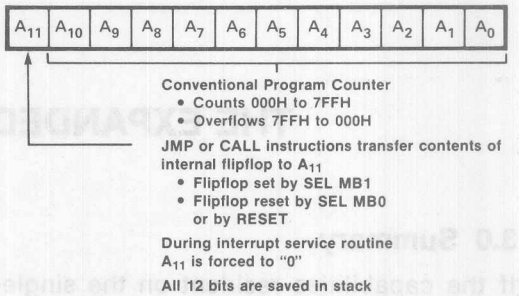
the conventional manner. Addresses beyond 2047 can be reached by executing a program memory bank switch instruction (SEL MB0, SEL MB1) followed by a branch instruction (JMP or CALL). The bank switch feature extends the range of branch instructions beyond their normal 2K range and at the same time prevents the user from inadvertently crossing the 2K boundary.

Program Memory Bank Switch

The switching of 2K program memory banks is accomplished by directly setting or resetting the most significant bit of the program counter (bit 11). Bit 11 is not altered by normal incrementing of the program counter but is loaded with the contents of a special flip-flop each time a JMP or CALL instruction is executed. This special flip-flop is set by executing an SEL MB1 instruction and reset by SEL MB0. Therefore, the SEL MB instruction may be executed at any time prior to the actual bank switch which occurs during the next branch instruction encountered. Since all twelve bits of the program counter including bit (11) are stored in the stack when a Call is executed, the user may jump to subroutines across the 2K boundary and the proper bank will be restored upon return. However, the bank switch flipflop will not be altered on return.

Interrupt Routines

Interrupts always vector the program counter to location 3 or 7 in the **first** 2K bank and bit 11 of the program counter is held at "0" during the interrupt service routine. The end of the service routine is signalled by the execution of an RETR instruction. Interrupt service routines should therefore be contained



PROGRAM COUNTER

entirely in the lower 2K words of program memory. The execution of a SEL MB0 or SEL MB1 instruction within an interrupt routine is not recommended since it will not alter PC11 while in the routine, but will change the internal flip flop.

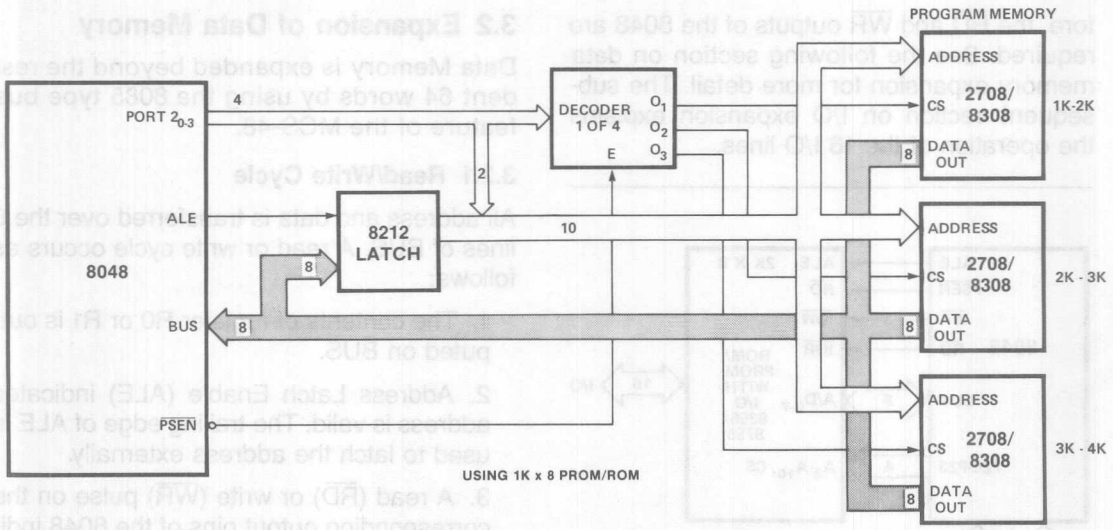
3.1.3 Restoring I/O Port Information

Although the lower half of Port 2 is used to output the four most significant bits of address during an external program memory fetch, the I/O information is still outputed during certain portions of each machine cycle. I/O information is always present on Port 2 lower at the rising edge of ALE and can be sampled or latched at this time.

3.1.4 Expansion Examples

The accompanying figure shows the addition of three 2708 1K X 8 EPROMs or three 8308 pin-compatible ROM replacements for a total of 4K words of program memory. The BUS port of the 8048 is connected directly to the data output lines of the memories. The lower 8 bits of address are latched in an 8212 8-bit latch using ALE as the strobe. The lower half of Port 2 provides the upper 4 bits of address and since these address bits are stable for the duration of the program memory fetch, they do not have to be latched. Two of the upper address bits are connected directly to the address inputs of the memories while the two most significant bits are decoded to provide the three chip selects needed. The $\overline{\text{PSEN}}$ output of the 8048/8748 is used to enable the chip select lines and therefore the memories.

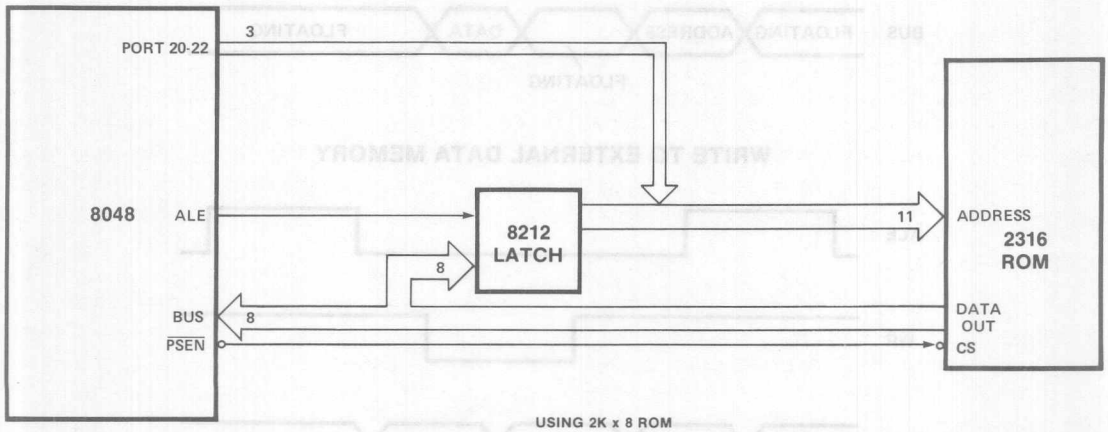
EXPANDED MCS-48 SYSTEM



EXPANDING MCS-48™ PROGRAM MEMORY USING STANDARD MEMORY PRODUCTS

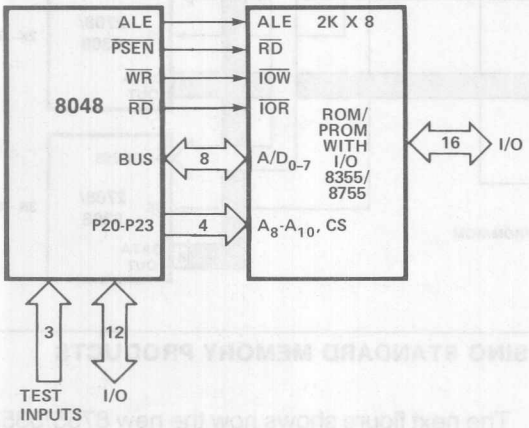
Also shown is the addition of 2K words of program memory using an 8316A 2K x 8 ROM to give a total of 3K words of program memory. In this case no chip select decoding is required and $\overline{PSEN}$ enables the memory directly through the chip select input. If the system requires only 2K of program the same configuration can be used with an 8035 substituted for the 8048. The 8049 would provide 4K with the same configuration.

The next figure shows how the new 8755/8355 EPROM/ROM with I/O interfaces directly to the 8048 without the need for an address latch. The 8755/8355 contains an internal 8-bit address latch eliminating the need for an 8212 latch. In addition to a 2K X 8 program memory the 8755/8355 also contains 16 I/O lines addressable as two 8-bit ports. These ports are addressed as external RAM; there-



EXPANDING MCS-48™ PROGRAM MEMORY USING STANDARD MEMORY PRODUCTS

fore, the $\overline{RD}$ and $\overline{WR}$ outputs of the 8048 are required. See the following section on data memory expansion for more detail. The subsequent section on I/O expansion explains the operation of the 16 I/O lines.



EXTERNAL PROGRAM MEMORY INTERFACE

3.2 Expansion of Data Memory

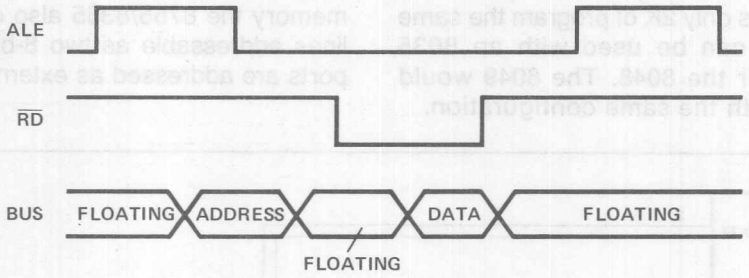
Data Memory is expanded beyond the resi dent 64 words by using the 8085 type bus feature of the MCS-48.

3.2.1 Read/Write Cycle

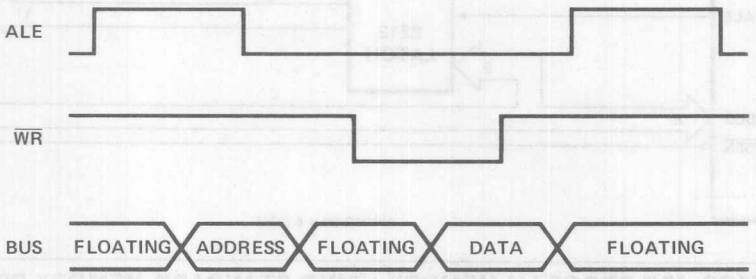
All address and data is transferred over the 8 lines of BUS. A read or write cycle occurs as follows:

1. The contents of register R0 or R1 is outputed on BUS.
2. Address Latch Enable (ALE) indicates address is valid. The trailing edge of ALE is used to latch the address externally.
3. A read ($\overline{RD}$) or write ($\overline{WR}$) pulse on the corresponding output pins of the 8048 indicates the type of data memory access in progress. Output data is valid at the trailing edge of $\overline{WR}$ and input data must be valid at the trailing edge of $\overline{RD}$.
4. Data (8-bits) is transferred in or out over BUS.

READ FROM EXTERNAL DATA MEMORY



WRITE TO EXTERNAL DATA MEMORY



3.2.2 Addressing External Data Memory

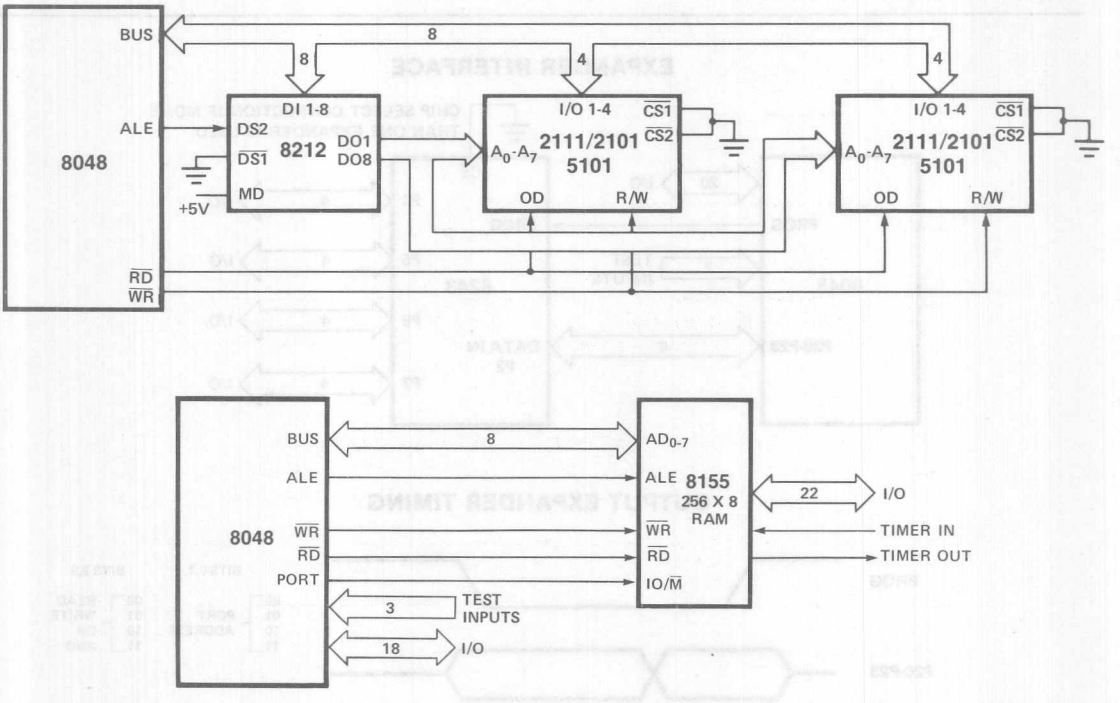
External Data Memory is accessed with its own two-cycle move instructions MOVX A, @R and MOVX @R, A which transfer 8 bits of data between the accumulator and the external memory location addressed by the contents of one of the RAM Pointer Registers R0 or R1. This allows 256 locations to be addressed in addition to the resident locations. Additional pages may be added by "bank switching" with extra output lines of the 8048.

3.2.3 Examples of Data Memory Expansion

The accompanying figure shows how the 8048 can be expanded using standard 256 X 4 static RAMs such as the 2101-2 or its low power CMOS equivalent, the 5101. An 8212 serves as an address latch while each 4-bit half of BUS is connected directly to a bidirectional 4-bit data bus of the memories. The $\overline{WR}$ output of the processor controls the Read/Write input of the memories while the data bus output drivers of the memories are controlled by $\overline{RD}$. The chip select lines of the memories are continuously enabled unless additional pages of RAM are required. Also shown is the expansion of data memory using the 8155 memory and I/O expanding device. Since the 8155 has an internal 8-bit address latch it can interface directly to the 8048 without the use of an external 8212 latch. The 8155 provides an additional 256 words of static data memory and also includes 22 I/O lines and a 14 bit timer. See the following section on I/O expansion and the 8155 data sheet for more details on these additional features.

3.3 Expansion of Input/Output

There are four possible modes of I/O expansion with the 8048: one using a special low cost expander, the 8243;



8048 INTERFACE TO 256 X 8 STANDARD MEMORIES

another using standard MCS-80/85 I/O devices; and a third using the combination memory/I/O expander devices the 8155, 8355, and 8755. It is also possible to expand using standard TTL devices as shown in Chapter 5.

3.3.1 I/O Expander Device

The most efficient means of I/O expansion for small systems is the 8243 I/O Expander Device which requires only 4 port lines (lower half of Port 2) for communication with the 8048. The 8243 contains four 4-bit I/O ports which serve as extension of the on chip I/O and are addressed as ports #4-7. The following operations may be performed on these ports:

1. Transfer Accumulator to Port.
2. Transfer Port to Accumulator.
3. AND Accumulator to Port.
4. OR Accumulator to Port.

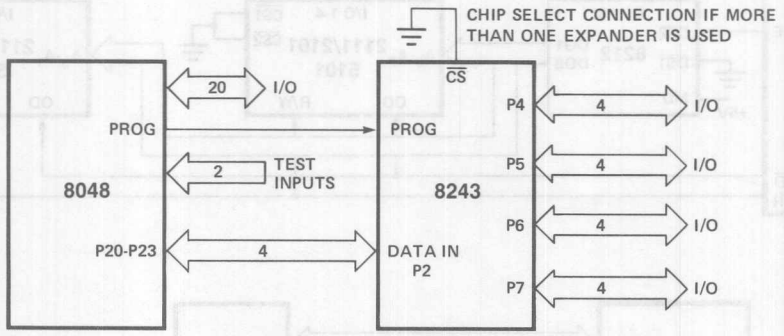
A 4-bit transfer from a port to the lower half of the Accumulator sets the most significant four bits to zero. All communication between the 8048 and the 8243 occurs over Port 2 lower (P20-P23) with timing provided by an output pulse on the PROG pin of the processor. Each transfer consists of two 4-bit nibbles:

The first containing the "op code" and port address and the second containing the actual 4 bits of data.

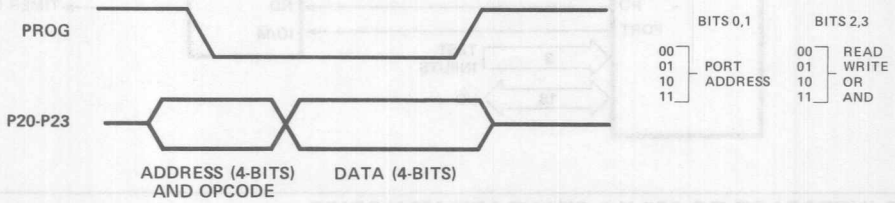
Nibble 1				Nibble 2			
3	2	1	0	3	2	1	0
I	I	A	A	d	d	d	d
Instruction Code		Port Address		data			

II	AA
00 Read	00—Port #4
01 Write	01—Port #5
10 OR	10—Port #6
11 AND	11—Port #7

EXPANDER INTERFACE



OUTPUT EXPANDER TIMING



A high to low transition of the PROG line indicates that address is present while a low to high transition indicates the presence of data. Additional 8243's may be added to the four bit bus and chip selected using additional output lines from the 8048/8748.

I/O Port Characteristics

Each of the four 4-bit ports of the 8243 can serve as either input or output and can provide high drive capability in both the high and low state.

3.3.2 I/O Expansion with Standard Peripherals

Standard MCS-80/85 type I/O devices may be added to the MCS-48 using the same bus and timing used for Data Memory expansion. I/O devices reside on the Data Memory bus and in the data memory address space and are accessed with the same MOVX instructions. See the previous section on data memory expansion for a description of timing. The following are a few of the Standard MCS-80 devices which are very useful in MCS-48 systems.

8214 Priority Interrupt Encoder
8251 Serial Communications Interface
8255 General Purpose Programmable I/O
8279 Keyboard/Display Interface
8253 Interval Timer

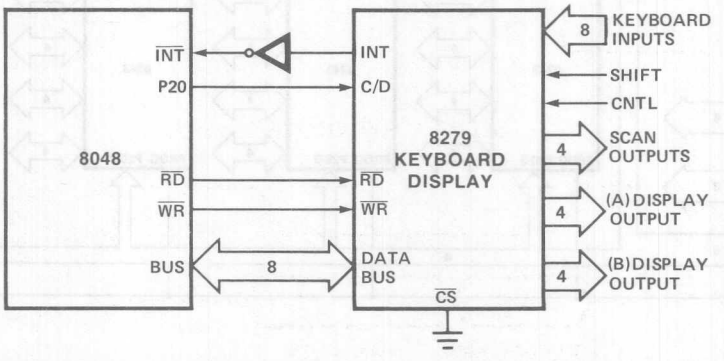
See Chapter 7 for detailed data sheets on these and other components.

3.3.3 Combination Memory and I/O Expanders

As mentioned in the sections on program and data memory expansion the 8355/8755 and 8155 expanders also contain I/O capability.

8355/8755: These two parts are ROM and EPROM equivalents and therefore contain the same I/O structure. I/O consists of two 8-bit ports which normally reside in the external data memory address space and are accessed with MOVX instructions. Associated with each port is an 8-bit Data Direction Register which defines each bit in the port as either an input or an output. The data direction registers are directly addressable thereby allowing the user to define under software control each individual bit of the ports as either input or output. All outputs are statically latched and double buffered. Inputs are not latched.

8155/8156: I/O on the 8155/8156 is configured as two 8-bit programmable I/O ports and one 6-bit programmable port. These three registers and a Control/Status register are accessible as external data memory with the MOVX instructions. The contents of the control register determines the mode of the three ports. The ports can be programmed as input or output with or without associated handshake communication lines. In the handshake mode, lines of the six-bit port become input and output strobes for the two 8-bit ports. See the



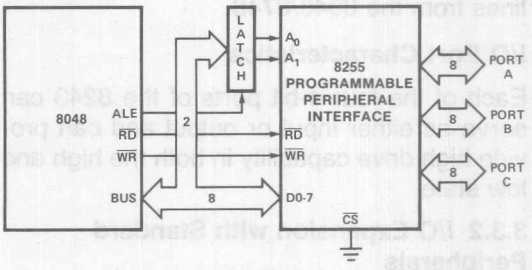
KEYBOARD/DISPLAY INTERFACE

data sheet in the Chapter 6 for details. Also included in the 8155 is a 14-bit programmable timer. The clock input to the timer and the timer overflow output are available on external pins. The timer can be programmed to stop on terminal count or to continuously reload itself. A square wave or pulse output on terminal count can also be specified.

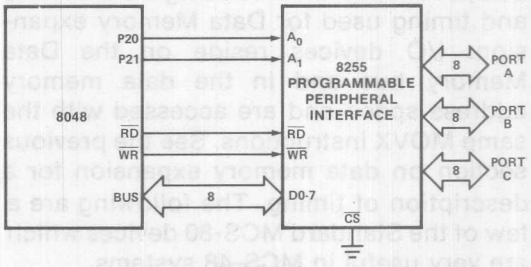
I/O Expansion Examples (See Also Chapter 5)

The accompanying figure shows the expansion of I/O using multiple 8243's. The only difference from a single 8243 system is the addition of chip selects provided by additional 8048 output lines. Two output lines and two inverters could also be used to address the four chips. Large numbers of 8243's would require a chip select decoder chip such as the 8205 to save I/O pins.

Also shown is the 8048 interface to a standard MCS-80 peripheral; in this case, the 8255 Programmable Peripheral Interface, a 40 pin part which provides three 8-bit programmable I/O ports. The 8255 bus interface is typical of programmable MCS-80 peripherals with an 8-bit bidirectional data bus, a $\overline{RD}$ and $\overline{WR}$ input for Read/Write control, a $\overline{CS}$ (chip select) input used to enable the Read/Write control logic and the address inputs used to select various internal registers.

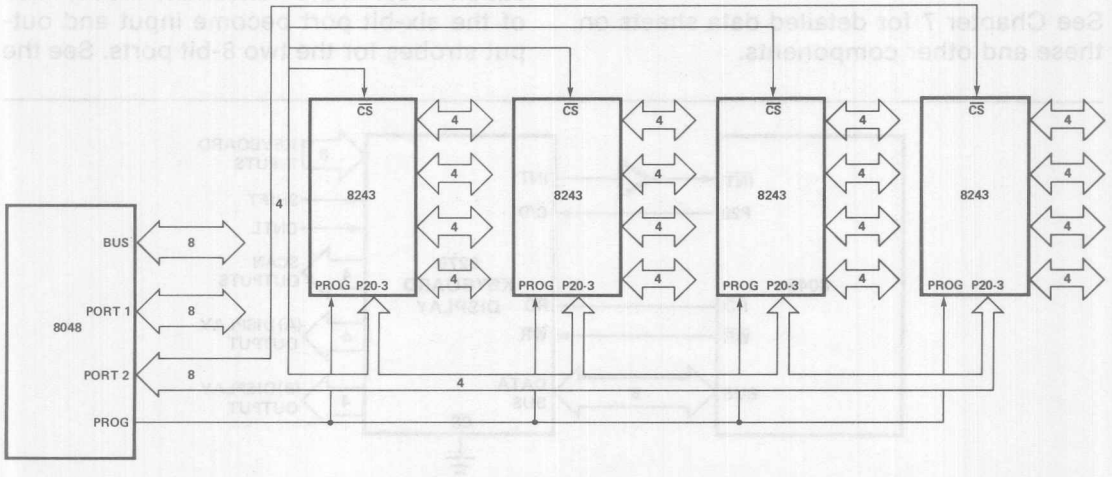


OPTION #1



OPTION #2

INTERFACE TO MCS-80 PERIPHERALS



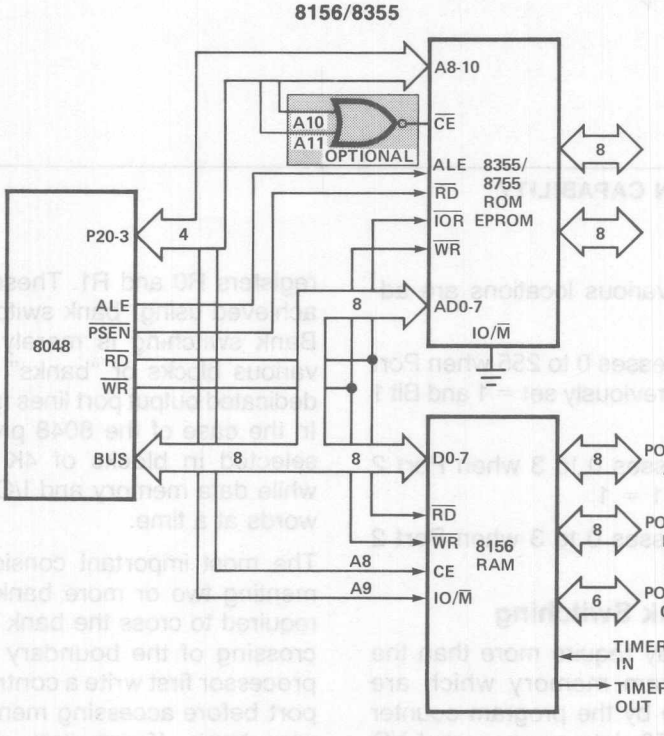
LOW COST I/O EXPANSION

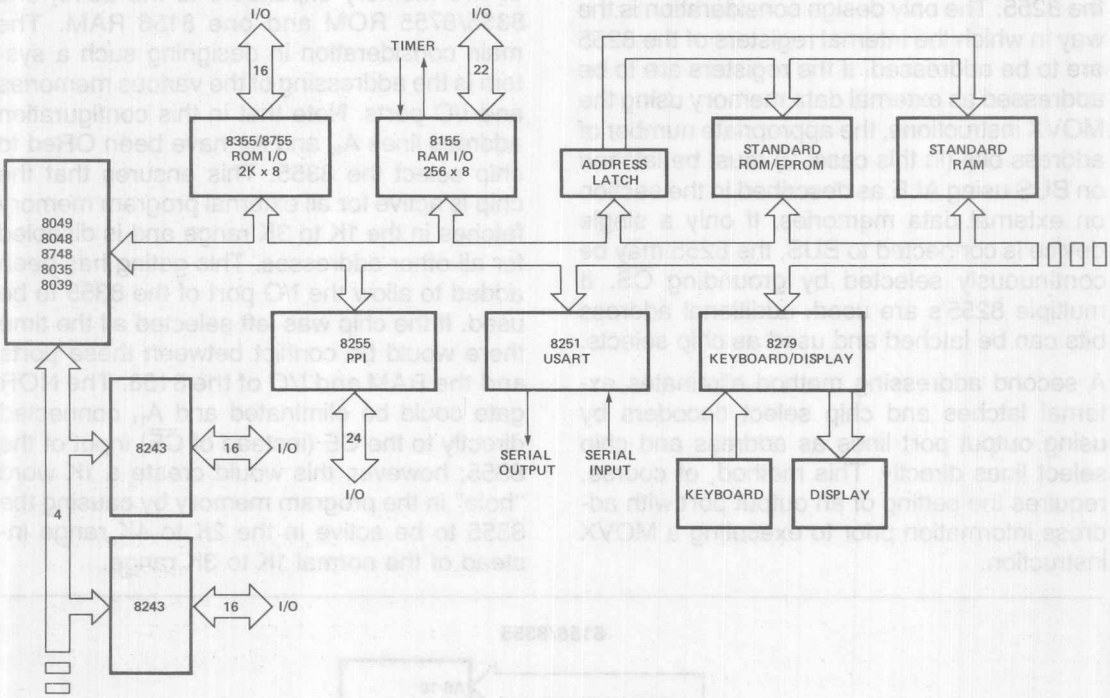
interconnection to the 8048 is very straightforward with BUS, $\overline{RD}$, and $\overline{WR}$ connecting directly to the corresponding pins on the 8255. The only design consideration is the way in which the internal registers of the 8255 are to be addressed. If the registers are to be addressed as external data memory using the MOVX instructions, the appropriate number of address bits (in this case, 2) must be latched on BUS using ALE as described in the section on external data memories. If only a single device is connected to BUS, the 8255 may be continuously selected by grounding $\overline{CS}$. If multiple 8255's are used, additional address bits can be latched and used as chip selects.

A second addressing method eliminates external latches and chip select decoders by using output port lines as address and chip select lines directly. This method, of course, requires the setting of an output port with address information prior to executing a MOVX instruction.

3.4 Multi-Chip MCS-48 Systems

The accompanying figure shows the addition of two memory expanders to the 8048, one 8355/8755 ROM and one 8156 RAM. The main consideration in designing such a system is the addressing of the various memories and I/O ports. Note that in this configuration address lines A_{10} and A_{11} have been ORed to chip select the 8355. This ensures that the chip is active for all external program memory fetches in the 1K to 3K range and is disabled for all other addresses. This gating has been added to allow the I/O port of the 8355 to be used. If the chip was left selected all the time there would be conflict between these ports and the RAM and I/O of the 8156. The NOR gate could be eliminated and A_{11} connected directly to the CE (instead of $\overline{CE}$) input of the 8355; however, this would create a 1K word "hole" in the program memory by causing the 8355 to be active in the 2K to 4K range instead of the normal 1K to 3K range.





MCS-48 EXPANSION CAPABILITY

In this system the various locations are addressed as follows:

Data RAM—Addresses 0 to 255 when Port 2 Bit 0 has been previously set = 1 and Bit 1 set = 0

RAM I/O—Addresses 0 to 3 when Port 2 Bit 0 = 1 and Bit 1 = 1

ROM I/O—Addresses 0 to 3 when Port 2 Bit 2 or Bit 3 = 1

3.5 Memory Bank Switching

Certain systems may require more than the 4K words of program memory which are directly addressable by the program counter or more than the 256 data memory and I/O locations directly addressable by the pointer registers R0 and R1. These systems can be achieved using "bank switching" techniques. Bank switching is merely the selection of various blocks or "banks" of memory using dedicated output port lines from the processor. In the case of the 8048 program memory is selected in blocks of 4K words at a time while data memory and I/O are enabled 256 words at a time.

The most important consideration in implementing two or more banks is the software required to cross the bank boundaries. Each crossing of the boundary requires that the processor first write a control bit to an output port before accessing memory or I/O in the new bank. If program memory is being switched, programs should be organized to

keep boundary crossings to a minimum. Jumping to subroutines across the boundary should be avoided when possible since the programmer must keep track of which bank to return to after completion of the subroutine. If these subroutines are to be nested and accessed from either bank, a software "stack" should be implemented to save the bank switch bit just as if it were another bit of the program counter.

From a hardware standpoint bank switching is very straight-forward and involves only the connection of an I/O line or lines as bank enable signals. These enables are ANDed with normal memory and I/O chip select signals to activate the proper bank.

3.6 Control Signal Summary

The following table summarizes the instructions which activate the various control outputs of the MCS-48 processors.

CONTROL SIGNAL	WHEN ACTIVE
$\overline{RD}$	DURING MOVX A,@R OR INS BUS
$\overline{WR}$	DURING MOVX @R,A OR OUTL BUS
ALE	EVERY MACHINE CYCLE
$\overline{PSEN}$	DURING FETCH OF EXTERNAL PROGRAM MEMORY (INSTRUCTION OR IMMEDIATE DATA)
PROG	DURING MOVD A,P ANLD P,A MOVD P,A ORLD P,A

During all other instructions these outputs are driven to the inactive state.

3.7 Port Characteristics

BUS Port Operations

The BUS port can operate in three different modes: as a latched I/O port, as a bi-directional bus port, or as a program memory address output when external memory is used. The BUS port lines are either active high, active low, or high impedance (floating). The latched mode (!NS, OUTL) is intended for use in the single chip configuration where BUS is not being used as an expander port. OUTL and MOVX instructions can be mixed if necessary. However, a previously latched output will be destroyed by executing a MOVX instruction and BUS will be left in the high impedance state. OUTL should never be used in a system with external program memory, since latching BUS can cause the next instruction, if external, to be fetched improperly.

Port 2 Operations

The lower half of Port 2 can be used in three different ways: as a quasi, bi-directional static port, as an 8243 expander port, and to address external program memory. In all cases outputs are driven low by an active device and driven high momentarily by an active device and held high by a 50KΩ resistor to +5V.

The port may contain latched I/O data prior to its use in another mode without affecting operation of either. If lower Port 2 (P20-3) is used to output address for an external program memory fetch the I/O information previously latched will be automatically removed temporarily while address is present then restored when the fetch is complete. However, if lower Port 2 is used to communicate with an 8243, previously latched I/O information will be removed and not restored. After an input from the 8243 P_{20-3} will be left in the input mode (floating). After an output to the 8243 P_{20-3} will contain the value written, ANDed, or ORed to the 8243 port.

Chapter 4
INSTRUCTION SET



INSTRUCTION SET

4.0 Introduction 4-1
4.1 Instruction Set Description 4-4

INSTRUCTION SET

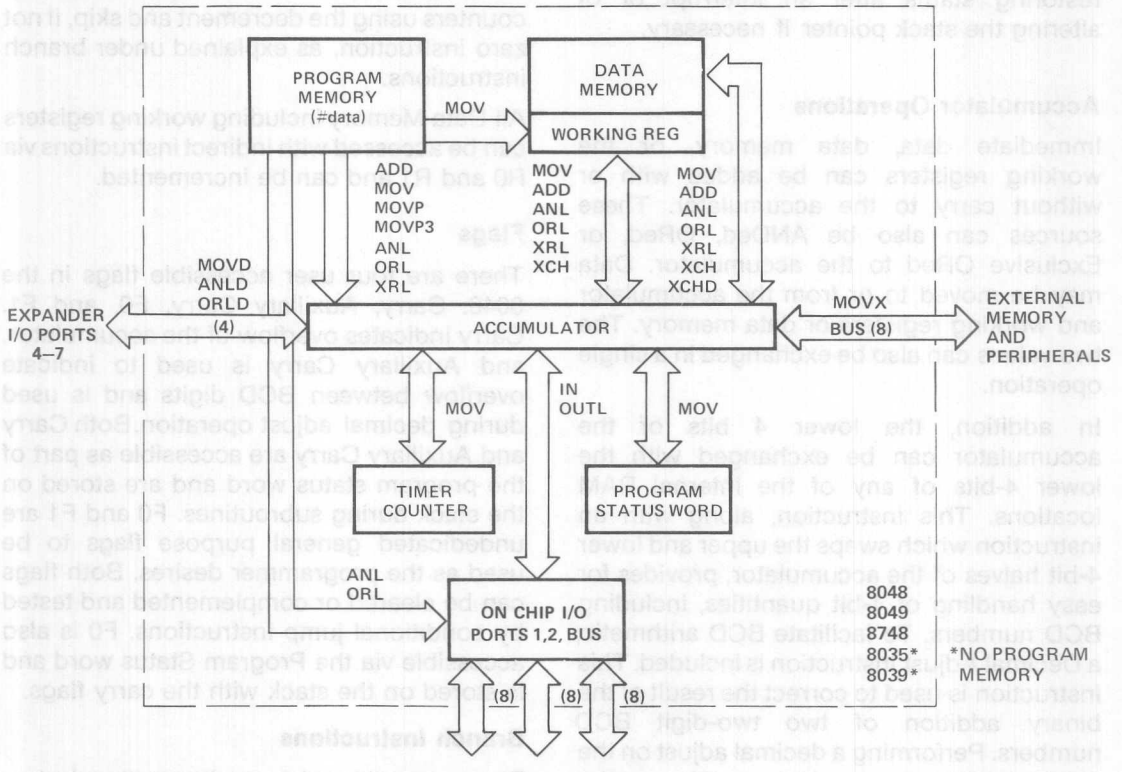
4.0 INTRODUCTION

The MCS-48 instruction set is extensive for a machine of its size and has been tailored to be straightforward and very efficient in its use of program memory. All instructions are either one or two bytes in length and over 70% are only one byte long. Also, all instructions execute in either one or two cycles (2.5μsec or 5.0μsec when using a 6 MHz XTAL) and over 50% of all instructions execute in a single cycle. Double cycle instructions include all immediate instructions, and all I/O instructions.

The MCS-48 microcomputers have been designed to efficiently handle arithmetic operations in both binary and BCD as well as to efficiently handle the single bit operations required in control applications. Special instructions have also been included to simplify loop counters, table lookup routines, and N-way branch routines.

Data Transfers

As can be seen in the accompanying diagram, the 8-bit accumulator is the central



DATA TRANSFER INSTRUCTIONS

point for all data transfers within the 8048. Data can be transferred between the 8 registers of each working register bank and the accumulator directly, i.e. the source or destination register is specified by the instruction. The remaining locations of the internal RAM array are referred to as Data Memory and are addressed indirectly via an address stored in either R0 or R1 of the active working register bank. R0 and R1 are also used to indirectly address external data memory when it is present. Transfers to and from internal RAM require one cycle while transfers to external RAM require two. Constants stored in Program Memory can be loaded directly to the accumulator and to the 8 working registers. Data can also be transfered directly between the accumulator and the on-board timer/counter or the accumulator and the Program Status word (PSW). Writing to the PSW alters machine status accordingly and provides a means of restoring status after an interrupt or of altering the stack pointer if necessary.

Accumulator Operations

Immediate data, data memory, or the working registers can be added with or without carry to the accumulator. These sources can also be ANDed, ORed, or Exclusive ORed to the accumulator. Data may be moved to or from the accumulator and working registers or data memory. The two values can also be exchanged in a single operation.

In addition, the lower 4 bits of the accumulator can be exchanged with the lower 4-bits of any of the internal RAM locations. This instruction, along with an instruction which swaps the upper and lower 4-bit halves of the accumulator, provides for easy handling of 4-bit quantities, including BCD numbers. To facilitate BCD arithmetic, a Decimal Adjust instruction is included. This instruction is used to correct the result of the binary addition of two two-digit BCD numbers. Performing a decimal adjust on the result in the accumulator produces the required BCD result.

Finally, the accumulator can be: incremented. decremented, cleared, or complemented and can be rotated left or right 1-bit at a time with or without carry.

Although there is no subtract instruction in the 8048, this operation can be easily implemented with three single-byte single-cycle instructions.

A value may be subtracted from the accumulator with the result in the accumulator by:

Complementing the accumulator
Adding the value to the accumulator
Complementing the accumulator.

Register Operations

The working registers can be accessed via the accumulator as explained above, or can be loaded immediate with constraints from program memory. In addition, they can be incremented or decremented or used as loop counters using the decrement and skip, if not zero instruction, as explained under branch instructions.

All Data Memory including working registers can be accessed with indirect instructions via R0 and R1 and can be incremented.

Flags

There are four user accessible flags in the 8048: Carry, Auxillary Carry, F0, and F1. Carry indicates overflow of the accumulator, and Auxillary Carry is used to indicate overflow between BCD digits and is used during decimal adjust operation. Both Carry and Auxillary Carry are accessible as part of the program status word and are stored on the stack during subroutines. F0 and F1 are undedicated general purpose flags to be used as the programmer desires. Both flags can be cleared or complemented and tested by conditional jump instructions. F0 is also accessible via the Program Status word and is stored on the stack with the carry flags.

Branch Instructions

The unconditional jump instruction is two bytes and allows jumps anywhere in the first

2K words of program memory. Jumps to the second 2K of memory (4K words are directly addressible) are made by first executing a select memory bank instruction then executing the jump instruction. The 2K boundary can only be crossed via a jump or subroutine call instruction i.e. the bank switch does not occur until a jump is executed. Once a memory bank has been selected all subsequent jumps will be to the selected bank until another select memory bank instruction is executed. A subroutine in the opposite bank can be accessed by a select memory bank instruction followed by a call instruction. Upon completion of the subroutine execution will automatically return to the original bank; however, unless the original bank is reselected, the next jump instruction encountered will again transfer execution to the opposite bank.

Conditional jumps can test the following inputs and machine status:

T0 Input pin
T1 Input pin
$\overline{INT}$ Input pin
Accumulator Zero
Any bit of Accumulator
Carry Flag
F0 Flag
F1 Flag

Conditional jumps allow a branch to any address within the current page (256 words) of execution. The conditions tested are the instantaneous values at the time the conditional jump is executed. For instance, the jump on accumulator zero instruction tests the accumulator itself not an intermediate zero flag.

The decrement register and skip if not zero instruction combines a decrement and a branch instruction to create an instruction very useful in implementing a loop counter. This instruction can designate any one of the 8 working registers as a counter and can effect a branch to any address within the current page of execution.

A single byte indirect jump instruction allows the program to be vectored to any one of several different locations based on the contents of the accumulator. The contents of the accumulator points to a location in program memory which contains the jump address. The 8-bit jump address refers to the current page of execution. This instruction could be used, for instance, to vector to any one of several routines based on an ASCII character which has been loaded in the accumulator. In this way ASCII key inputs can be used to initiate various routines.

Subroutines

Subroutines are entered by executing a call instruction. Calls can be made like unconditional jumps to any address in a 2K word bank and jumps across the 2K boundary are executed in the same manner. Two separate return instructions determine whether or not status (upper 4-bits of PSW) is restored upon return from the subroutine.

The return and restore status instruction also signals the end of an interrupt service routine if one has been in progress.

Timer Instructions

The 8-bit on board timer/counter can be loaded or read via the accumulator while the counter is stopped or while counting. The counter can be started as a timer with an internal clock source or as an event counter or timer with an external clock applied to the T1 input pin. The instruction executed determines which clock source is used. A single instruction stops the counter whether it is operating with an internal or an external clock source. In addition, two instructions allow the timer interrupt to be enabled or disabled.

Control Instructions

Two instructions allow the external interrupt source to be enabled or disabled. Interrupts are initially disabled and are automatically disabled while an interrupt service routine is in progress and re-enabled afterward.

There are four memory bank select instructions, two to designate the active working register bank and two to control program

memory banks. The operation of the program memory bank switch is explained in section 3.1.2. The working register bank switch instructions allow the programmer to immediately substitute a second 8 register working register bank for the one in use. This effectively provides 16 working registers or it can be used as a means of quickly saving the contents of the registers in response to an interrupt. The user has the option to switch or not to switch banks on interrupt. However, if the banks are switched, the original bank will be automatically restored upon execution of a return and restore status instruction at the end of the interrupt service routine.

A special instruction enables an internal clock, which is the XTAL frequency divided by three, to be output on pin T0. This clock can be used as a general purpose clock in the users system. This instruction should be used only to initialize the system since the clock output can be disabled only by application of system reset.

Input/Output Instructions

Ports 1 and 2 are 8-bit static I/O ports which can be loaded to and from the accumulator. Outputs are statically latched but inputs are not latched and must be read while inputs are present. In addition, immediate data from program memory can be ANDed or ORed directly to Port 1 and Port 2 with the result remaining on the port. This allows "masks" stored in program memory to selectively set or reset individual bits of the I/O ports. Ports 1 and 2 are configured to allow input on a given pin by first writing a "1" out to the pin.

An 8-bit port called BUS can also be accessed via the accumulator and can have statically latched outputs as well. It too can have immediate data ANDed or ORed directly to its outputs, however, unlike ports 1 and 2, all eight lines of BUS must be treated as either input or output at any one time. In addition to being a static port, BUS can be used as a true synchronous bi-directional port using the Move External instructions used to access external data memory. When these instructions are executed a corresponding READ or WRITE pulse is generated and data is valid only at that time. When data is not being transferred BUS is in a high impedance state.

The basic three on board I/O ports can be expanded via a 4-bit expander bus using half of port 2. I/O expander devices on this bus consist of four 4-bit ports which are addressed as ports 4 through 7. These ports have their own AND and OR instructions like the on board ports as well as move instructions to transfer data in or out. The expander AND and OR instructions, however, combine the contents of accumulator with the selected port rather than immediate data as is done with the on board ports.

I/O devices can also be added externally using the BUS port as the expansion bus. In this case the I/O ports become "memory mapped", i.e. they are addressed in the same way as external data memory and exist in the external data memory address space addressed by pointer register R0 or R1.

4.1 Instruction Set Description

The following pages describe the MCS-48 instruction set in detail. The instruction set is first summarized with instructions grouped functionally. This summary page is followed by a detailed description listed alphabetically by mnemonic opcode.

The alphabetical listing includes the following information:

Mnemonic
Machine Code
Verbal Description
Symbolic Description
Assembly Language Example

The machine code is represented with the most significant bit (7) to the left and two byte instructions are represented with the first byte on the left. The assembly language examples are formulated as follows:

```
Arbitrary
Label:  Mnemonic, Operand;      Descriptive Comment
```

See section 1.2.2 for a description and example of an assembly language program.

8048/8049
INSTRUCTION SET SUMMARY

	Mnemonic	Description	Bytes	Cycle
Accumulator	ADD A, R	Add register to A	1	1
	ADD A, @R	Add data memory to A	1	1
	ADD A, #data	Add immediate to A	2	2
	ADDC A, R	Add register with carry	1	1
	ADDC A, @R	Add data memory with carry	1	1
	ADDC A, #data	Add immediate with carry	2	2
	ANL A, R	And register to A	1	1
	ANL A, @R	And data memory to A	1	1
	ANL A, #data	And immediate to A	2	2
	ORL A, R	Or register to A	1	1
	ORL A, @R	Or data memory to A	1	1
	ORL A, #data	Or immediate to A	2	2
	XRL A, R	Exclusive Or register to A	1	1
	XRL A, @R	Exclusive or data memory to A	1	1
	XRL A, #data	Exclusive or immediate to A	2	2
	INC A	Increment A	1	1
	DEC A	Decrement A	1	1
	CLR A	Clear A	1	1
	CPL A	Complement A	1	1
	DA A	Decimal Adjust A	1	1
	SWAP A	Swap nibbles of A	1	1
	RL A	Rotate A left	1	1
	RLC A	Rotate A left through carry	1	1
	RR A	Rotate A right	1	1
	RRC A	Rotate A right through carry	1	1
Input/Output	IN A, P	Input port to A	1	2
	OUTL P, A	Output A to port	1	2
	ANL P, #data	And immediate to port	2	2
	ORL P, #data	Or immediate to port	2	2
	INS A, BUS	Input BUS to A	1	2
	OUTL BUS, A	Output A to BUS	1	2
	ANL BUS, #data	And immediate to BUS	2	2
	ORL BUS, #data	Or immediate to BUS	2	2
	MOVD A, P	Input Expander port to A	1	2
	MOVD P, A	Output A to Expander port	1	2
	ANLD P, A	And A to Expander port	1	2
	ORLD P, A	Or A to Expander port	1	2
Registers	INC R	Increment register	1	1
	INC @R	Increment data memory	1	1
	DEC R	Decrement register	1	1
Branch	JMP addr	Jump unconditional	2	2
	JMPP @A	Jump indirect	1	2
	DJNZ R, addr	Decrement register and skip	2	2
	JC addr	Jump on Carry = 1	2	2
	JNC addr	Jump on Carry = 0	2	2
	J Z addr	Jump on A Zero	2	2
	JNZ addr	Jump on A not Zero	2	2
	JT0 addr	Jump on T0 = 1	2	2
	JNT0 addr	Jump on T0 = 0	2	2
	JT1 addr	Jump on T1 = 1	2	2
	JNT1 addr	Jump on T1 = 0	2	2
	JF0 addr	Jump on F0 = 1	2	2
	JF1 addr	Jump on F1 = 1	2	2
	JTF addr	Jump on timer flag	2	2
	JNI addr	Jump on $\overline{INT}$ = 0	2	2
	JBb addr	Jump on Accumulator Bit	2	2

	Mnemonic	Description	Bytes	Cycles
Subroutine	CALL	Jump to subroutine	2	2
	RET	Return	1	2
	RETR	Return and restore status	1	2
Flags	CLR C	Clear Carry	1	1
	CPL C	Complement Carry	1	1
	CLR F0	Clear Flag 0	1	1
	CPL F0	Complement Flag 0	1	1
	CLR F1	Clear Flag 1	1	1
	CPL F1	Complement Flag 1	1	1
Data Moves	MOV A, R	Move register to A	1	1
	MOV A, @R	Move data memory to A	1	1
	MOV A, #data	Move immediate to A	2	2
	MOV R, A	Move A to register	1	1
	MOV @R, A	Move A to data memory	1	1
	MOV R, #data	Move immediate to register	2	2
	MOV @R, #data	Move immediate to data memory	2	2
	MOV A, PSW	Move PSW to A	1	1
	MOV PSW, A	Move A to PSW	1	1
	XCH A, R	Exchange A and register	1	1
	XCH A, @R	Exchange A and data memory	1	1
	XCHD A, @R	Exchange nibble of A and register	1	1
	MOVX A, @R	Move external data memory to A	1	2
	MOVX @R, A	Move A to external data memory	1	2
	MOVP A, @A	Move to A from current page	1	2
	MOVP3 A, @A	Move to A from Page 3	1	2
Timer/Counter	MOV A, T	Read Timer/Counter	1	1
	MOV T, A	Load Timer/Counter	1	1
	STRT T	Start Timer	1	1
	STRT CNT	Start Counter	1	1
	STOP TCNT	Stop Timer/Counter	1	1
	EN TCNTI	Enable Timer/Counter Interrupt	1	1
	DIS TCNTI	Disable Timer/Counter Interrupt	1	1
Control	EN I	Enable external interrupt	1	1
	DIS I	Disable external interrupt	1	1
	SEL RB0	Select register bank 0	1	1
	SEL RB1	Select register bank 1	1	1
	SEL MB0	Select memory bank 0	1	1
	SEL MB1	Select memory bank 1	1	1
	ENT0 CLK	Enable Clock output on T0	1	1
	NOP	No Operation	1	1

Mnemonics copyright Intel Corporation 1976.

8021
INSTRUCTION SET SUMMARY

	Mnemonic		Description	Bytes	Cycle
Accumulator	ADD	A,R	Add register to A	1	1
	ADD	A,@R	Add data memory to A	1	1
	ADD	A,#data	Add immediate to A	2	2
	ADDC	A,R	Add with carry	1	1
	ADDC	A,@R	Add with carry	1	1
	ADDC	A,#data	Add with carry	2	2
	ANL	A,R	And register to A	1	1
	ANL	A,@R	And data memory to A	1	1
	ANL	A,#data	And immediate to A	2	2
	ORL	A,R	Or register to A	1	1
	ORL	A,@R	Or data memory to A	1	1
	ORL	A,#data	Or immediate to A	2	2
	XRL	A,R	Exclusive Or register to A	1	1
	XRL	A,@R	Exclusive or data memory to A	1	1
	XRL	A,#data	Exclusive or immediate to A	2	2
	INC	A	Increment A	1	1
	DEC	A	Decrement A	1	1
	CLR	A	Clear A	1	1
	CPL	A	Complement A	1	1
	DA	A	Decimal Adjust A	1	1
	SWAP	A	Swap nibbles of A	1	1
	RL	A	Rotate A left	1	1
	RLC	A	Rotate A left through carry	1	1
	RR	A	Rotate A right	1	1
	RRC	A	Rotate A right through carry	1	1
Input/Output	IN	A,P	Input port to A	1	2
	OUTL	P,A	Output A to port	1	2
	MOVD	A,P	Input Expander port to A	1	2
	MOVD	P,A	Output A to Expander port	1	2
	ANLD	P,A	And A to Expander port	1	2
	ORLD	P,A	Or A to Expander port	1	2
Registers	INC	R	Increment register	1	1
	INC	@R	Increment data memory	1	1

	Mnemonic		Description	Bytes	Cycle
Branch	JMP	addr	Jump unconditional	2	2
	JMPP	@A	Jump indirect	1	2
	DJNZ	R,addr	Decrement register and Jump on R not zero	2	2
	JC	addr	Jump on Carry = 1	2	2
	JNC	addr	Jump on Carry = 0	2	2
	JZ	addr	Jump on A Zero	2	2
	JNZ	addr	Jump on A not Zero	2	2
	JT1	addr	Jump on T1 = 1	2	2
	JNT1	addr	Jump on T1 = 0	2	2
	JTF	addr	Jump on timer flag	2	2
Subroutine	CALL		Jump to subroutine	2	2
	RET		Return	1	2
Flags	CLR	C	Clear Carry	1	1
	CPL	C	Complement Carry	1	1
Data Moves	MOV	A,R	Move register to A	1	1
	MOV	A,@R	Move data memory to A	1	1
	MOV	A,#data	Move immediate to A	2	2
	MOV	R,A	Move A to register	1	1
	MOV	@R,A	Move A to data memory	1	1
	MOV	R,#data	Move immediate to register	2	2
	MOV	@R,#data	Move immediate to data memory	2	2
	XCH	A,R	Exchange A and register	1	1
	XCH	A,@R	Exchange A and data memory	1	2
	XCHD	A,@R	Exchange nibble of A and register	1	1
	MOVP	A,@A	Move to A from current page	1	2
Timer/Counter	MOV	A,T	Read Timer/Counter	1	1
	MOV	T,A	Load Timer/Counter	1	1
	STRT	T	Start Timer	1	1
	STRT	CNT	Start Counter	1	1
	STOP	TCNT	Stop Timer/Counter	1	1
	NOP		No Operation	1	1

Instruction Set — The following instructions, which are found in the 8748, have been deleted from the 8021 instruction set.

Data Moves	Registers	Branch	Timer	Control	Input/Output
MOV A,PSW	DEC R	JT0 addr	EN TCNTI	EN I	ANL P,#data
MOV PSW,A		JNT0 addr	DIS TCNTI	DIS I	ORL P,#data
MOVX A,@R	**Flags**	JF0 addr		SEL RB0	INS A,BUS *
MOVX @R,A	CLR F0	JF1 addr	**Subroutine**	SEL RB1	OUTL BUS,A *
MOVP3 A,@A	CPL F0	JNI addr	RETR	SEL MB0	ANL BUS,#data
	CLR F1	JBb addr		SEL MB1	ORL BUS,#data
	CPL F1			ENT0 CLK	

*These Instructions have been replaced in the 8021 by IN A,P0 and OUTL P0,A respectively.

MCS-48™ INSTRUCTION SET

SYMBOLS AND ABBREVIATIONS USED

A	Accumulator
AC	Auxillary Carry
addr	12-Bit Program Memory Address
Bb	Bit Designator (b=0-7)
BS	Bank Switch
BUS	BUS Port
C	Carry
CLK	Clock
CNT	Event Counter
D	Mnemonic for 4-Bit Digit (Nibble)
data	8-Bit Number or Expression
DBF	Memory Bank Flip-Flop
F0, F1	Flag 0, Flag 1
I	Interrupt
P	Mnemonic for "in-page" Operation
PC	Program Counter
Pp	Port Designator (p=1, 2 or 4-7)
PSW	Program Status Word
Rr	Register Designator (r=0, 1 or 0-7)
SP	Stack Pointer
T	Timer
TF	Timer Flag
T0, T1	Test 0, Test 1
X	Mnemonic for External RAM
#	Immediate Data Prefix
@	Indirect Address Prefix
$	Current Value of Program Counter
(X)	Contents of X
((X))	Contents of Location Addressed by X
←	Is Replaced by

Mnemonics copyright Intel Corporation 1976.

ADD A,R_r Add Register Contents to Accumulator

0 1 1 0	1 r r r

The contents of register 'r' are added to the accumulator. Carry is affected.

(A) ← (A) + (Rr) r=0-7

Example:
```
ADDREG: ADD A,R6    ;ADD REG 6 CONTENTS
                    ;TO ACC
```

ADD A,@R_r Add Data Memory Contents to Accumulator

0 1 1 0	0 0 0 r

The contents of the resident data memory location addressed by register 'r' bits 0-5*are added to the accumulator. Carry is affected.

(A) ← (A) + ((Rr)) r=0-1

Example:
```
ADDM: MOV R0, #01FH  ;MOVE '1F' HEX TO REG 0
      ADD A, @R0     ;ADD VALUE OF LOCATION
                     ;47 TO ACC
```

ADD A,#data Add Immediate Data to Accumulator

0 0 0 0	0 0 1 1	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The specified data is added to the accumulator. Carry is affected

(A) ← (A) + data

Example:
```
ADDID: ADD A,#ADDER: ;ADD VALUE OF SYMBOL
                     ;'ADDER' TO ACC
```

ADDC A,R_r Add Carry and Register Contents to Accumulator

0 1 1 1	1 r r r

The content of the carry bit is added to accumulator location 0 and the carry bit cleared. The contents of register 'r' are then added to the accumulator. Carry is affected.

(A) ← (A)+(Rr)+(C) r=0-7

Example:
```
ADDRGC: ADDC A,R4   ;ADD CARRY AND REG 4
                    ;CONTENTS TO ACC
```

Mnemonics copyright Intel Corporation 1976.

*0-6 for 8039/8049

ADDC A,@R_r Add Carry and Data Memory Contents to Accumulator

0 1 1 1	0 0 0 r

The content of the carry bit is added to accumulator location 0 and the carry bit cleared. Then the contents of the resident data memory location addressed by register 'r' bits 0-5*are added to the accumulator. Carry is affected.

(A) ← (A)+((Rr))+(C) r=0-1

Example:
```
ADDMC: MOV R1,#40    ;MOVE '40' DEC TO REG 1
       ADDC A,@R1    ;ADD CARRY AND LOCATION 40
                     ;CONTENTS TO ACC
```

ADDC A,#data Add Carry and Immediate Data to Accumulator

0 0 0 1	0 0 1 1	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The content of the carry bit is added to accumulator location 0 and the carry bit cleared. Then the specified data is added to the accumulator. Carry is affected.

(A) ← (A)+data+(C)

Example:
```
ADDC A,#225      ;ADD CARRY AND '225' DEC
                 ;TO ACC
```

ANL A,R_r Logical AND Accumulator With Register Mask

0 1 0 1	1 r r r

Data in the accumulator is logically ANDed with the mask contained in working register 'r'.

(A) ← (A) AND (Rr) r=0-7

Example:
```
ANDREG: ANL A,R3    ;'AND' ACC CONTENTS WITH MASK
                    ;IN REG 3
```

ANL A,@R_r Logical AND Accumulator With Memory Mask

0 1 0 1	0 0 0 r

Data in the accumulator is logically ANDed with the mask contained in the data memory location referenced by register 'r', bits 0-5*.

(A) ← (A) AND ((Rr)) r=0-1

Example:
```
ANDDM: MOV R0,#03FH ;MOVE '3F' HEX TO REG 0
       ANL A, @R0   ;'AND' ACC CONTENTS WITH
                    ;MASK IN LOCATION 63
```

Mnemonics copyright Intel Corporation 1976.

*0-6 for 8039/8049

ANL A,#data Logical AND Accumulator With Immediate Mask

0 1 0 1	0 0 1 1	$d_7\ d_6\ d_5\ d_4$	$d_3\ d_2\ d_1\ d_0$

This is a 2-cycle instruction. Data in the accumulator is logically ANDed with an immediately-specified mask.

(A) ← (A) AND data

Examples:
```
ANDID: ANL A,#0AFH    ;'AND' ACC CONTENTS
                      ;WITH MASK 10101111
       ANL A,#3+X/Y   ;'AND' ACC CONTENTS
                      ;WITH VALUE OF EXP
                      ;'3+X/Y'
```

ANL BUS,#data Logical AND BUS With Immediate Mask (Not in 8021)

1 0 0 1	1 0 0 0	$d_7\ d_6\ d_5\ d_4$	$d_3\ d_2\ d_1\ d_0$

This is a 2-cycle instruction. Data on the BUS port is logically ANDed with an immediately-specified mask. This instruction assumes prior specification of an 'OUTL BUS, A' instruction.

(BUS) ← (BUS) AND data

Example:
```
ANDBUS: ANL BUS, #MASK ;'AND' BUS CONTENTS
                       ;WITH MASK EQUAL VALUE
                       ;OF SYMBOL 'MASK'
```

ANL Pp,#data Logical AND Port 1-2 With Immediate Mask (Not in 8021)

1 0 0 1	1 0 p p	$d_7\ d_6\ d_5\ d_4$	$d_3\ d_2\ d_1\ d_0$

This is a 2-cycle instruction. Data on port 'p' is logically ANDed with an immediately-specified mask.

(Pp) ← (Pp) AND data p=1-2

Example:
```
ANDP2: ANL P2,#0F0H   ;'AND' PORT 2 CONTENTS
                      ;WITH MASK 'F0' HEX
                      ;(CLEAR P20-23)
```

ANLD Pp,A Logical AND Port 4-7 With Accumulator Mask

1 0 0 1	1 1 p p

This is a 2-cycle instruction. Data on port 'p' is logically ANDed with the digit mask contained in accumulator bits 0-3.

(Pp) ← (Pp) AND (A0-3) p=4-7

Mnemonics copyright Intel Corporation 1976.

Note: The mapping of port 'p' to opcode bits 0-1 is as follows:

1 0	Port
0 0	4
0 1	5
1 0	6
1 1	7

Example:
```
ANDP4: ANLD P4,A        ;'AND' PORT 4 CONTENTS
                        ;WITH ACC BITS 0-3
```

CALL address Subroutine Call

a_{10} a_9 a_8 1	0 1 0 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. The program counter and PSW bits 4-7 are saved in the stack. The stack pointer (PSW bits 0-2) is updated. Program control is then passed to the location specified by 'address'. PC bit 11 is determined by the most recent SEL MB instruction.

Execution continues at the instruction following the CALL upon return from the subroutine.

((SP)) — (PC), (PSW_{4-7})
(SP) — (SP)+1
(PC_{8-10}) — ($addr_{8-10}$)
(PC_{0-7}) — $addr_{0-7}$
(PC_{11}) — DBF

Example: Add three groups of two numbers. Put subtotals in locations 50, 51 and total in location 52.

```
        MOV R0,#50      ;MOVE '50' DEC T0 ADDRESS
                        ;REG 0
BEGADD: MOV A,R1        ;MOVE CONTENTS OF REG 1
                        ;TO ACC
        ADD A,R2        ;ADD REG 2 TO ACC
        CALL SUBTOT     ;CALL SUBROUTINE 'SUBTOT'
        ADD A R3        ;ADD REG 3 TO ACC
        ADD A,R4        ;ADD REG 4 TO ACC
        CALL SUBTOT     ;CALL SUBROUTINE 'SUBTOT'
        ADD A,R5        ;ADD REG 5 TO ACC
        ADD A,R6        ;ADD REG 6 TO ACC
        CALL SUBTOT     ;CALL SUBROUTINE 'SUBTOT'

SUBTOT: MOV @R0,A       ;MOVE CONTENTS OF ACC TO
                        ;LOCATION ADDRESSED BY
                        ;REG 0
        INC R0          ;INCREMENT REG 0
        RET             ;RETURN TO MAIN PROGRAM
```

Mnemonics copyright Intel Corporation 1976.

CLR A Clear Accumulator

0 0 1 0	0 1 1 1

The contents of the accumulator are cleared to zero.

A ← 0

CLR C Clear Carry Bit

1 0 0 1	0 1 1 1

During normal program execution, the carry bit can be set to one by the ADD, ADDC, RLC, CPL C, RRC, and DAA instructions. This instruction resets the carry bit to zero.

C ← 0

CLR F1 Clear Flag 1 (Not in 8021)

1 0 1 0	0 1 0 1

Flag 1 is cleared to zero.

(F1) ← 0

CLR F0 Clear Flag 0 (Not in 8021)

1 0 0 0	0 1 0 1

Flag 0 is cleared to zero.

(F0) ← 0

CPL A Complement Accumulator

0 0 1 1	0 1 1 1

The contents of the accumulator are complemented. This is strictly a one's complement. Each one is changed to zero and vice-versa.

(A) ← NOT (A)

Example: Assume accumulator contains 01101010.

```
CPLA: CPL A          ;ACC CONTENTS ARE COMPLE-
                     ;MENTED TO 10010101
```

CPL C Complement Carry Bit

1 0 1 0	0 1 1 1

The setting of the carry bit is complemented; one is changed to zero, and zero is changed to one.

(C) ← NOT (C)

Example: Set C to one; current setting is unknown.

```
CTO1: CLR C          ;C IS CLEARED TO ZERO
      CPL C          ;C IS SET TO ONE
```

Mnemonics copyright Intel Corporation 1976.

CPL F0 Complement Flag 0 (Not in 8021)

1 0 0 1	0 1 0 1

The setting of flag 0 is complemented; one is changed to zero, and zero is changed to one.

F0← NOT (F0)

CPL F1 Complement Flag 1 (Not in 8021)

1 0 1 1	0 1 0 1

The setting of flag 1 is complemented; one is changed to zero, and zero is changed to one.

(F1)← NOT (F1)

DA A Decimal Adjust Accumulator

0 1 0 1	0 1 1 1

The 8-bit accumulator value is adjusted to form two 4-bit Binary Coded Decimal (BCD) digits following the binary addition of BCD numbers. The carry bit C is affected. If the contents of bits 0-3 are greater than nine, or if AC is one, the accumulator is incremented by six.

The four high-order bits are then checked. If bits 4-7 exceed nine, or if C is one, these bits are increased by six. If an overflow occurs, C is set to one.

Example: Assume accumulator contains 10011011.

```
DA A                    ;ACC ADJUSTED TO 00000001
                        ;WITH C SET
C  AC  7   4  3   0
0  0   1 0 0 1  1 0 1 1
                0 1 1 0     ADD SIX TO BITS 0-7
0  0    1 0 1 0  0 0 0 1
        0 1 1 0             ADD SIX TO BITS 4-7
1  0    0 0 0 0  0 0 0 1     OVERFLOW TO C
```

DEC A Decrement Accumulator

0 0 0 0	0 1 1 1

The contents of the accumulator are decremented by one.

(A)← (A)-1

Mnemonics copyright Intel Corporation 1976.

Example: Decrement contents of external data memory location 63.

```
MOV R0,#3FH      ;MOVE '3F' HEX TO REG 0
MOVX A,@R0       ;MOVE CONTENTS OF LOCATION 63
                 ;TO ACC
DEC A            ;DECREMENT ACC
MOVX @R0,A       ;MOVE CONTENTS OF ACC TO
                 ;LOCATION 63 IN EXPANDED
                 ;MEMORY
```

DEC R_r Decrement Register (Not in 8021)

1 1 0 0	1 r r r

The contents of working register 'r' are decremented by one.

$(Rr) \leftarrow (Rr)-1$ r=0-7

Example:
```
DECR1: DEC R1    ;DECREMENT CONTENTS OF REG 1
```

DIS I Disable External Interrupt (Not in 8021)

0 0 0 1	0 1 0 1

External interrupts are disabled. A low signal on the interrupt input pin has no effect.

DIS TCNTI Disable Timer/Counter Interrupt (Not in 8021)

0 0 1 1	0 1 0 1

Timer/counter interrupts are disabled. Any pending timer interrupt request is cleared. The interrupt sequence is not initiated by an overflow, but the timer flag is set and time accumulation continues.

DJNZ R_r, address Decrement Register and Test

1 1 1 0	1 r r r	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Register 'r' is decremented and tested for zero. If the register contains all zeros, program control falls through to the next instruction. If the register contents are not zero, control jumps to the specified 'address'.

The address in this case must evaluate to 8-bits, that is, the jump must be to a location within the current 256-location page.

$(Rr) \leftarrow (Rr)-1$ r=0-7
If Rr not 0
$(PC_{0-7}) \leftarrow$ addr

Mnemonics copyright Intel Corporation 1976.

Note: A 12-bit address specification does not cause an error if the DJNZ instruction and the jump target are on the same page. If the DJNZ instruction begins in location 255 of a page, it must jump to a target address on the following page.

Example: Increment values in data memory locations 50-54.

```
        MOV R0,#50      ;MOVE '50' DEC TO ADDRESS
                        ;REG 0
        MOV R3,#5       ;MOVE '5' DEC TO COUNTER
                        ;REG 3
INCRT:  INC @R0         ;INCREMENT CONTENTS OF
                        ;LOCATION ADDRESSED BY
                        ;REG 0
        INC R0          ;INCREMENT ADDRESS IN REG 0
        DJNZ R3, INCRT  ;DECREMENT REG 3 — JUMP TO
                        ;'INCRT' IF REG 3 NONZERO
        NEXT —          ;'NEXT' ROUTINE EXECUTED
                        ;IF R3 IS ZERO
```

EN I Enable External Interrupt (Not in 8021)

0 0 0 0	0 1 0 1

External interrupts are enabled. A low signal on the interrupt input pin initiates the interrupt sequence.

EN TCNTI Enable Timer/Counter Interrupt (Not in 8021)

0 0 1 0	0 1 0 1

Timer/counter interrupts are enabled. An overflow of the timer/counter initiates the interrupt sequence.

ENT0 CLK Enable Clock Output (Not in 8021)

0 1 1 1	0 1 0 1

The test 0 pin is enabled to act as the clock output. This function is disabled by a system reset.

Example:

```
EMTST0: ENT0 CLK        ;ENABLE T0 AS CLOCK OUTPUT
```

IN A,Pp Input Port or Data to Accumulator

0 0 0 0	1 0 p p

This is a 2-cycle instruction. Data present on port 'p' is transferred (read) to the accumulator. In the 8021 IN A,P2 inputs P_{20}-P_{23} to A_0-A_3 while A_4-A_7 is set to zero.

(A) ← (Pp) p=1-2

Mnemonics copyright Intel Corporation 1976.

```
Example: INP12: IN A,P1          ;INPUT PORT 1 CONTENTS
                                 ;TO ACC
         MOV R6,A                ;MOVE ACC CONTENTS TO
                                 ;REG 6
         IN A,P2                 ;INPUT PORT 2 CONTENTS
                                 ;TO ACC
         MOV R7,A                ;MOVE ACC CONTENTS TO REG 7
```

INC A Increment Accumulator

0 0 0 1	0 1 1 1

The contents of the accumulator are incremented by one.

(A) ← (A)+1

Example: Increment contents of location 100 in external data memory.

```
INCA: MOV R0,#100        ;MOVE '100' DEC TO ADDRESS
                         ;REG 0
      MOVX A,@R0         ;MOVE CONTENTS OF LOCATION
                         ;100 TO ACC
      INC A              ;INCREMENT A
      MOVX @R0,A         ;MOVE ACC CONTENTS TO
                         ;LOCATION 100
```

INC R_r Increment Register

0 0 0 1	1 r r r

The contents of working register 'r' are incremented by one.

(Rr) ← (Rr)+1 r=0-7

```
Example: INCR0: INC R0           ;INCREMENT ADDRESS REG 0
```

INC @R_r Increment Data Memory Location

0 0 0 1	0 0 0 r

The contents of the resident data memory location addressed by register 'r' bits 0-5*are incremented by one.

((Rr)) ← ((Rr))+1 r=0-1

```
Example: INCDM: MOV R1,#03FH     ;MOVE ONES TO REG 1
             INC @R1             ;INCREMENT LOCATION 63
```

Mnemonics copyright Intel Corporation 1976.

*0-6 for 8039/8049

IN A,P0 Input of Port 0 Data to Accumulator

Same as INS A,BUS except no $\overline{RD}$ pulse generated.

INS A,BUS Strobed Input of BUS Data to Accumulator

0 0 0 0	1 0 0 0

This is a 2-cycle instruction. Data present on the BUS port is transferred (read) to the accumulator when the $\overline{RD}$ pulse is dropped. (Refer to section on programming memory expansion for details).

(A)← (BUS)

Example:
```
INPBUS: INS A,BUS      ;INPUT BUS CONTENTS
                       ;TO ACC
```

JBb address Jump If Accumulator Bit is Set (Not in 8021)

$b_2\ b_1\ b_0\ 1$	0 0 1 0	$a_7\ a_6\ a_5\ a_4$	$a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control passes to the specified address if accumulator bit 'b' is set to one.

(PC_{0-7})← addr	If Bb=1
(PC) = (PC)+2	If Bb=0

Example:
```
JB4IS1: JB4 NEXT       ;JUMP TO 'NEXT' ROUTINE
                       ;IF ACC BIT 4=1
```

JC address Jump If Carry Is Set

1 1 1 1	0 1 1 0	$a_7\ a_6\ a_5\ a_4$	$a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control passes to the specified address if the carry bit is set to one.

(PC_{0-7})← addr	If C=1
(PC) = (PC)+2	If C=0

Example:
```
JC1: JC OVFLOW         ;JUMP TO 'OVFLOW' ROUTINE
                       ;IF C=1
```

JF0 address Jump If Flag 0 Is Set (Not in 8021)

1 0 1 1	0 1 1 0	$a_7\ a_6\ a_5\ a_4$	$a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control passes to the specified address if flag 0 is set to one.

(PC_{0-7})← addr	If F0=1
(PC) = (PC)+2	If F0=0

Example:
```
JF0IS1: JF0 TOTAL      ;JUMP TO 'TOTAL' ROUTINE
                       ;IF F0=1
```

Mnemonics copyright Intel Corporation 1976.

JF1 address Jump If Flag 1 Is Set (Not in 8021)

0 1 1 1	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if flag 1 is set to one.

$(PC_{0-7}) \leftarrow$ addr If F1=1
(PC) = (PC)+2 IF F1=0

Example:
```
JF1IS1: JF1 FILBUF    ;JUMP TO 'FILBUF'
                      ;ROUTINE IF F1=1
```

JMP address Direct Jump Within 2K Block

a_{10} a_9 a_8 0	0 1 0 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Bits 0-10 of the program counter are replaced with the directly-specified address. The setting of PC bit 11 is determined by the most recent SELECT MB instruction.

$(PC_{8-10}) \leftarrow$ addr 8-10
$(PC_{0-7}) \leftarrow$ addr 0-7
$(PC_{11}) \leftarrow$ DBF

Example:
```
JMP SUBTOT    ;JUMP TO SUBROUTINE 'SUBTOT'
JMP $-6       ;JUMP TO INSTRUCTION SIX LOCATIONS
              ;BEFORE CURRENT LOCATION
JMP 2FH       ;JUMP TO ADDRESS '2F' HEX
```

JMPP @A Indirect Jump Within Page

1 0 1 1	0 0 1 1

This is a 2-cycle instruction. The contents of the program memory location pointed to by the accumulator are substituted for the 'page' portion of the program counter (PC bits 0-7).

$(PC_{0-7}) \leftarrow ((A))$

Example: Assume accumulator contains 0FH.
```
JMPPAG: JMPP @A    ;JUMP TO ADDRESS STORED IN
                   ;LOCATION 15 IN CURRENT PAGE
```

JNC address Jump If Carry Is Not Set

1 1 1 0	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the carry bit is not set, that is, equals zero.

Mnemonics copyright Intel Corporation 1976.

	$(PC_{0-7}) \leftarrow$ addr	If C=0
	(PC) = (PC)+2	IF C=1
Example:	JC0: JNC NOVFLO	;JUMP TO 'NOVFLO' ROUTINE ;If C=0

JNI address Jump If Interrupt Input is Low (Not in 8021)

1 0 0 0	0 1 1 0	$a_7\ a_6\ a_5\ a_4$	$a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control passes to the specified address if the interrupt input signal is low (=0), that is, an external interrupt has been signaled. (This signal initiates an interrupt service sequence if the external interrupt is enabled.)

	$(PC_{0-7}) \leftarrow$ addr	If I=0
	(PC) = (PC)+2	If I=1
Example:	LOC 3: JNI EXTINT	;JUMP TO 'EXTINT' ROUTINE ;If I=0

JNT0 address Jump If Test 0 Is Low (Not in 8021)

0 0 1 0	0 1 1 0	$a_7\ a_6\ a_5\ a_4$	$a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control passes to the specified address, if the test 0 signal is low

	$(PC_{0-7}) \leftarrow$ addr	If T0=0
	(PC) = (PC)+2	If T0=1
Example:	JT0LOW: JNT0 60	;JUMP TO LOCATION 60 DEC ;IF T0=0

JNT1 address Jump If Test 1 Is Low

0 1 0 0	0 1 1 0	$a_7\ a_6\ a_5\ a_4$	$a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control passes to the specified address, if the test 1 signal is low.

	$(PC_{0-7}) \leftarrow$ addr	If T1=0
	(PC) = (PC)+2	If T1=1

JNZ address Jump If Accumulator Is Not Zero

1 0 0 1	0 1 1 0	$a_7\ a_6\ a_5\ a_4$	$a_3\ a_2\ a_1\ a_0$

This is a 2-cycle instruction. Control pases to the specified address if the accumulator contents are nonzero at the time this instruction is executed.

	$(PC_{0-7}) \leftarrow$ addr	If A≠0
	(PC) = (PC)+2	If A=0
Example:	JACCN0: JNZ 0ABH	;JUMP TO LOCATION 'AB' HEX ;IF ACC VALUE IS NONZERO

Mnemonics copyright Intel Corporation 1976.

0 0 0 1	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the timer flag is set to one, that is, the timer/counter register has overflowed. Testing the timer flag resets it to zero. (This overflow initiates an interrupt service sequence if the timer-overflow interrupt is enabled.)

$(PC_{0-7}) \leftarrow$ addr	If TF=1
(PC) = (PC)+2	If TF=0

Example:
```
JTF1: JTF TIMER      ;JUMP TO 'TIMER' ROUTINE
                     ;IF TF=1
```

JT0 address Jump If Test 0 Is High (Not in 8021)

0 0 1 1	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the test 0 signal is high (=1).

$(PC_{0-7}) \leftarrow$ addr	If T0=1
(PC) = (PC)+2	If T0=0

Example:
```
JT0HI: JT0 53        ;JUMP TO LOCATION 53 DEC
                     ;IF T0=1
```

JT1 address Jump If Test 1 Is High

0 1 0 1	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the test 1 signal is high (=1).

$(PC_{0-7}) \leftarrow$ addr	If T1=1
(PC) = (PC)+2	If T1=0

Example:
```
JT1HI: JT1 COUNT     ;JUMP TO 'COUNT' ROUTINE
                     ;IF T1=1
```

JZ address Jump If Accumulator Is Zero

1 1 0 0	0 1 1 0	a_7 a_6 a_5 a_4	a_3 a_2 a_1 a_0

This is a 2-cycle instruction. Control passes to the specified address if the accumulator contains all zeros at the time this instruction is executed.

$(PC_{0-7}) \leftarrow$ addr	If A=0
(PC) = (PC)+2	If A≠0

Example:
```
JACCO: JZ 0A3H       ;JUMP TO LOCATION 'A3' HEX
                     ;IF ACC VALUE IS ZERO
```

Mnemonics copyright Intel Corporation 1976.

MOV A, #data Move Immediate Data to Accumulator

0 0 1 0	0 0 1 1	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The 8-bit value specified by 'data' is loaded in the accumulator.

(A) ← data

Example: MOV A,#0A3H ;MOVE 'A3' HEX TO ACC

MOV A,PSW Move PSW Contents to Accumulator (Not in 8021)

1 1 0 0	0 1 1 1

The contents of the program status word are moved to the accumulator.

(A) ← (PSW)

Example: Jump to 'RB1SET' routine if PSW bank switch, bit 4, is set.

```
BSCHK: MOV A,PSW    ;MOVE PSW CONTENTS TO ACC
       JB4 RB1SET   ;JUMP TO 'RB1SET' IF ACC
                    ;BIT 4=1
```

MOV A,R_r Move Register Contents to Accumulator

1 1 1 1	1 r r r

8-bits of data are moved from working register 'r' into the accumulator.

(A) ← (Rr) r=0-7

Example:

```
MAR: MOV A,R3    ;MOVE CONTENTS OF REG 3
                 ;TO ACC
```

MOV A,@R_r Move Data Memory Contents to Accumulator

1 1 1 1	0 0 0 r

The contents of the resident data memory location addressed by bits 0-5*of register 'r' are moved to the accumulator. Register 'r' contents are unaffected.

(A) ← ((Rr)) r=0-1

Example: Assume R1 contains 00110110.

```
MADM: MOV A,@R1    ;MOVE CONTENTS OF DATA MEM
                   ;LOCATION 54 TO ACC
```

Mnemonics copyright Intel Corporation 1976.

*0-6 for 8039/8049

MOV A,T Move Timer/Counter Contents to Accumulator

0 1 0 0	0 0 1 0

The contents of the timer/event-counter register are moved to the accumulator.

(A)← (T)

Example: Jump to "EXIT" routine when timer reaches '64', that is, when bit 6 set — assuming initialization 64,

```
TIMCHK: MOV A,T     ;MOVE TIMER CONTENTS TO
                    ;ACC
        JB6 EXIT    ;JUMP TO 'EXIT' IF ACC BIT
                    ;6=1
```

MOV PSW,A Move Accumulator Contents to PSW (Not in 8021)

1 1 0 1	0 1 1 1

The contents of the accumulator are moved into the program status word. All condition bits and the stack pointer are affected by this move.

(PSW)← (A)

Example: Move up stack pointer by two memory locations, that is, increment the pointer by one.

```
INCPTR: MOV A,PSW   ;MOVE PSW CONTENTS TO ACC
        INC A       ;INCREMENT ACC BY ONE
        MOV PSW,A   ;MOVE ACC CONTENTS TO PSW
```

MOV R_r,A Move Accumulator Contents to Register

1 0 1 0	1 r r r

The contents of the accumulator are moved to register 'r'.

(Rr)← (A) r=0-7

Example:

```
MRA: MOV R0,A       ;MOVE CONTENTS OF ACC TO
                    ;REG 0
```

MOV R_r,#data Move Immediate Data to Register

1 0 1 1	1 r_2 r_1 r_0	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to register 'r'.

(Rr)← data r=0-7

Mnemonics copyright Intel Corporation 1976.

Examples:
```
MIR4: MOV R4,#HEXTEN ;THE VALUE OF THE SYMBOL
                     ;'HEXTEN' IS MOVED INTO
                     ;REG 4
MIR 5: MOV R5,#PI*(R*R);THE VALUE OF THE
                     ;EXPRESSION 'PI*(R*R)
                     ;IS MOVED INTO REG 5
MIR 6: MOV R6, #0ADH ;'AD' HEX IS MOVED INTO
                     ;REG 6
```

MOV @R_r,A Move Accumulator Contents to Data Memory

1 0 1 0	0 0 0 r

This is a 2-cycle instruction. The contents of the accumulator are moved to the resident data memory location whose address is specified by bits 0-5* of register 'r'. Register 'r' contents are unaffected.

((Rr)) ← (A) r=0-1

Example: Assume R0 contains 00000111.

```
MDMA: MOV @R0,A      ;MOVE CONTENTS OF ACC TO
                     ;LOCATION 7 (REG 7)
```

MOV @R_r,#data Move Immediate Data to Data Memory

1 0 1 1	0 0 0 r	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. The 8-bit value specified by 'data' is moved to the resident data memory location addressed by register 'r', bits 0-5*.

((Rr)) ← data r=0-1

Examples: Move the hexadecimal value AC3F to locations 62-63.

```
MIDM: MOV R0,#62      ;MOVE '62' DEC TO ADDR REG 0
      MOV @R0,#0ACH   ;MOVE 'AC' HEX TO LOCATION 62
      INC R0          ;INCREMENT REG 0 TO '63'
      MOV @R0,#3FH    ;MOVE '3F' HEX TO LOCATION 63
```

MOV T,A Move Accumulator Contents to Timer/Counter

0 1 1 0	0 0 1 0

The contents of the accumulator are moved to the timer/event-counter register.

(T) ← (A)

Example: Initialize and start event counter.

```
INITEC: CLR A        ;CLEAR ACC TO ZEROS
        MOV T,A      ;MOVE ZEROS TO EVENT COUNTER
        STRT CNT     ;START COUNTER
```

Mnemonics copyright Intel Corporation 1976.

*0-6 for 8039/8049

MOVD A,Pp Move Port 4-7 Data to Accumulator

0 0 0 0	1 1 p p

This is a 2-cycle instruction. Data on 8243 port 'p' is moved (read) to accumulator bits 0-3. Accumulator bits 4-7 are zeroed.

(0-3) ← (Pp) p=4-7
(4-7) ← 0

Note: Bits 0-1 of the opcode are used to represent ports 4-7. If you are coding in binary rather than assembly language, the mapping is as follows:

Bits 1 0	Port
0 0	4
0 1	5
1 0	6
1 1	7

Example:
```
INPPT5: MOVD A,P5      ;MOVE PORT 5 DATA TO ACC
                       ;BITS 0-3, ZERO ACC BITS 4-7
```

MOVD Pp,A Move Accumulator Data to Port 4-7

0 0 1 1	1 1 p p

This is a 2-cycle instruction. Data in accumulator bits 0-3 is moved (written) to 8243 port 'p'. Accumulator bits 4-7 are unaffected. (See NOTE above regarding port mapping.)

$(Pp) \leftarrow (A_{0-3})$ p=4-7

Example: Move data in accumulator to ports 4 and 5.
```
OUTP45: MOVD P4,A      ;MOVE ACC BITS 0-3 TO PORT 4
        SWAP A         ;EXCHANGE ACC BITS 0-3 AND 4-7
        MOVD P5,A      ;MOVE ACC BITS 0-3 TO PORT 5
```

MOVP A,@A Move Current Page Data to Accumulator

1 0 1 0	0 0 1 1

The contents of the program memory location addressed by the accumulator are moved to the accumulator. Only bits 0-7 of the program counter are affected, limiting the program memory reference to the current page. The program counter is restored following this operation

$(PC_{0-7}) \leftarrow (A)$
$(A) \leftarrow ((PC))$

Note: This is a 1-byte, 2-cycle instruction. If it appears in location 255 of a program memory page, @A addresses a location in the following page.

Mnemonics copyright Intel Corporation 1976.

```
Example:  MOV128: MOV A,#128    ;MOVE '128' DEC TO ACC
                  MOVP A,@A     ;CONTENTS OF 129th LOCATION
                                ;IN CURRENT PAGE ARE MOVED TO
                                ;ACC
```

MOVP3 A,@A Move Page 3 Data to Accumulator (Not in 8021)

1 1 1 0	0 0 1 1

This is a 2-cycle instruction. The contents of the program memory location (within page 3) addressed by the accumulator are moved to the accumulator. The program counter is restored following this operation.

$(PC_{0-7}) \leftarrow (A)$
$(PC_{8-11}) \leftarrow 0011$
$(A) \leftarrow ((PC))$

Example: Look up ASCII equivalent of hexadecimal code in table contained at the beginning of page 3. Note that ASCII characters are designated by a 7-bit code; the eighth bit is always reset.

```
TABSCH: MOV A,#0B8H  ;MOVE 'B8' HEX TO ACC (10111000)
        ANL A,#7FH   ;LOGICAL AND ACC TO MASK BIT
                     ;7 (00111000)
        MOVP3 A,@A   ;MOVE CONTENTS OF LOCATION
                     ;'38' HEX IN PAGE 3 TO ACC
                     ;(ASCII '8')
```

Access contents of location in page 3 labelled TAB1. Assume current program location is not in page 3.

```
TABSCH: MOV A,#LOW TAB1 ;ISOLATE BITS 0-7 OF LABEL
                        ;ADDRESS VALUE
        MOVP3 A,@A      ;MOVE CONTENTS OF PAGE 3
                        ;LOCATION LABELED 'TAB1'
                        ;TO ACC
```

MOVX A,@R_r Move External-Data-Memory Contents to Accumulator

1 0 0 0	0 0 0 r

(Not in 8021)

This is a 2-cycle instruction. The contents of the external data memory location addressed by register 'r' are moved to the accumulator. Register 'r' contents are unaffected.

$(A) \leftarrow ((Rr))$ r=0-1

Example: Assume R1 contains 01110110.

```
MAXDM: MOVX A,@R1   ;MOVE CONTENTS OF LOCATION
                    ;118 TO ACC
```

Mnemonics copyright Intel Corporation 1976.

MOVX @R_r,A Move Accumulator Contents to External Data Memory

1 0 0 1	0 0 0 r

(Not in 8021)

This is a 2-cycle instruction. The contents of the accumulator are moved to the external data memory location addressed by register 'r'. Register 'r' contents are unaffected.

((Rr)) ← A

Example: Assume R0 contains 11000111.

```
MXDMA: MOVX @R0,A      ;MOVE CONTENTS OF ACC TO
                       ;LOCATION 199 IN EXPANDED
                       ;DATA MEMORY
```

NOP The NOP Instruction

0 0 0 0	0 0 0 0

No operation is performed. Execution continues with the following instruction.

ORL A,R_r Logical OR Accumulator With Register Mask

0 1 0 0	1 r r r

Data in the accumulator is logically ORed with the mask contained in working register 'r'.

(A) ← (A) OR (Rr) r=0-7

Example:

```
ORREG: ORL A,R4        ;'OR' ACC CONTENTS WITH
                       ;MASK IN REG 4
```

ORL A,@R_r Logical OR Accumulator With Memory Mask

0 1 0 0	0 0 0 r

Data in the accumulator is logically ORed with the mask contained in the resident data memory location referenced by register 'r', bits 0-5*

(A) ← (A) OR ((Rr)) r=0-1

Example:

```
ORDM: MOV R0,#3FH      ;MOVE '3F' HEX TO REG 0
      ORL A,@R0        ;'OR' ACC CONTENTS WITH MASK
                       ;IN LOCATION 63
```

ORL A,#data Logical OR Accumulator With Immediate Mask

0 1 0 0	0 0 1 1	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data in the accumulator is logically ORed with an immediately-specified mask.

(A) ← (A) OR data

Example:

```
ORID: ORL A,#'X'       ;'OR' ACC CONTENTS WITH MASK
                       ;01011000 (ASCII VALUE OF 'X')
```

Mnemonics copyright Intel Corporation 1976.

*0-6 for 8039/8049

ORL BUS,#data Logical OR BUS With Immediate Mask (Not in 8021)

1 0 0 0	1 0 0 0	$d_7\ d_6\ d_5\ d_4$	$d_3\ d_2\ d_1\ d_0$

This is a 2-cycle instruction. Data on the BUS port is logically ORed with an immediately-specified mask. This instruction assumes prior specification of an 'OUTL BUS,A' instruction.

(BUS)← (BUS) OR data

```
Example: ORBUS: ORL BUS,#HEXMSK  ;'OR' BUS CONTENTS WITH
                                 ;MASK EQUAL VALUE OF SYMBOL
                                 ;'HEXMSK'
```

ORL Pp, #data Logical OR Port 1 or 2 With Immediate Mask (Not in 8021)

1 0 0 0	1 0 p p	$d_7\ d_6\ d_5\ d_4$	$d_3\ d_2\ d_1\ d_0$

This is a 2-cycle instruction. Data on port 'p' is logically ORed with an immediately-specified mask.

(Pp)← (Pp) OR data p=1-2

```
Example: ORP1: ORL P1, #0FFH  ;'OR' PORT 1 CONTENTS WITH
                              ;MASK 'FF' HEX ( SET PORT 1
                              ;TO ALL ONES)
```

ORLD Pp,A Logical OR Port 4-7 With Accumulator Mask

1 0 0 0	1 1 p p

This is a 2-cycle instruction. Data on port 'p' is logically ORed with the digit mask contained in accumulator bits 0-3.

(Pp)← (Pp) OR $(A_{0\text{-}3})$ p=4-7

```
Example: ORP7: ORLD P7,A  ;'OR' PORT 7 CONTENTS
                          ;WITH ACC BITS 0-3
```

OUTL P0,A Output Accumulator Data to Port 0 (8021 only)

1 0 0 1	0 0 0 0

OUTL BUS,A Output Accumulator Data to BUS (Not in 8021)

0 0 0 0	0 0 1 0

This is a 2-cycle instruction. Data residing in the accumulator is transferred (written) to the BUS port and latched. The latched data remains valid until altered by another OUTL instruction. Any other instruction requiring use of the BUS port (except INS) destroys the contents of the BUS latch. This includes expanded memory operations (such as the MOVX instruction). Logical operations on BUS data (AND, OR) assume the OUTL BUS,A instruction has been issued previously.

Does not apply for OUTL P0,A of 8021

(BUS)← (A)

```
Example: OUTLBP: OUTL BUS,A  ;OUTPUT ACC CONTENTS TO BUS
```

Mnemonics copyright Intel Corporation 1976.

OUTL Pp,A Output Accumulator Data to Port 1 or 2

0 0 1 1	1 0 p p

This is a 2-cycle instruction. Data residing in the accumulator is transferred (written) to port 'p' and latched.

(Pp) ← (A) p=1-2

Example:
```
OUTLP: MOV A,R7     ;MOVE REG 7 CONTENTS TO ACC
       OUTL P2,A    ;OUTPUT ACC CONTENTS TO PORT 2
       MOV A,R6     ;MOVE REG 6 CONTENTS TO ACC
       OUTL P1,A    ;OUTPUT ACC CONTENTS TO PORT 1
```

RET Return Without PSW Restore

1 0 0 0	0 0 1 1

This is a 2-cycle instruction. The stack pointer (PSW bits 0-2) is decremented. The program counter is then restored from the stack. PSW bits 4-7 are not restored.

(SP) ← (SP)-1
(PC) ← ((SP))

RETR Return With PSW Restore (Not in 8021)

1 0 0 1	0 0 1 1

This is a 2-cycle instruction. The stack pointer is decremented. The program counter and bits 4-7 of the PSW are then restored from the stack. Note that RETR should be used to return from an interrupt, but should not be used within the interrupt service routine as it signals the end of an interrupt routine.

(SP) ← (SP)-1
(PC) ← ((SP))
(PSW 4-7) ← ((SP))

RL A Rotate Left Without Carry

1 1 1 0	0 1 1 1

The contents of the accumulator are rotated left one bit. Bit 7 is rotated into the bit 0 position.

(AN+1) ← (An)
(A0) ← (A7) n=0-6

Example: Assume accumulator contains 10110001.
```
RLNC: RL A          ;NEW ACC CONTENTS ARE 01100011.
```

Mnemonics copyright Intel Corporation 1976.

RLC A Rotate Left Through Carry

1 1 1 1	0 1 1 1

The contents of the accumulator are rotated left one bit. Bit 7 replaces the carry bit; the carry bit is rotated into the bit 0 position.

(AN+1) ← (An) n=0-6
(A0) ← (C)
(C) ← (A7)

Example: Assume accumulator contains a 'signed' number; isolate sign without changing value.

```
RLTC: CLR C      ;CLEAR CARRY TO ZERO
      RLC A      ;ROTATE ACC LEFT, SIGN
                 ;BIT (7) IS PLACED IN CARRY
      RR A       ;ROTATE ACC RIGHT — VALUE
                 (BITS 0-6) IS RESTORED,
                 ;CARRY UNCHANGED, BIT 7
                 ;IS ZERO
```

RR A Rotate Right Without Carry

0 1 1 1	0 1 1 1

The contents of the accumulator are rotated right one bit. Bit 0 is rotated into the bit 7 position

(An) ← (AN+1) n=0-6
(A7) ← (A0)

Example: Assume accumulator contains 10110001.

```
RRNC: RR A       ;NEW ACC CONTENTS ARE 11011000
```

RRC A Rotate Right Through Carry

0 1 1 0	0 1 1 1

The contents of the accumulator are rotated right one bit. Bit 0 replaces the carry bit; the carry bit is rotated into the bit 7 position.

(An) ← (An+1) n=0-6
(A7) ← (C)
(C) ←)A0)

Example: Assume carry is not set and accumulator contains 10110001.

```
RRTC: RRC A      ;CARRY IS SET AND ACC
                 ;CONTAINS 01011000
```

Mnemonics copyright Intel Corporation 1976.

SEL MBO Select Memory Bank 0 (Not in 8021)

1 1 1 0	0 1 0 1

PC bit 11 is set to zero on next JMP or CALL instruction. All references to program memory addresses fall within the range 0-2047.

(DBF) ← 0

Example: Assume program counter contains 834 Hex.

```
SEL MBO          ;SELECT MEMORY BANK 0
JMP $+20         ;JUMP TO LOCATION
                 ;48 HEX
```

SEL MB1 Select Memory Bank 1 (Not in 8021)

1 1 1 1	0 1 0 1

PC bit 11 is set to one on next JMP or CALL instruction. All references to program memory addresses fall within the range 2048-4095.

(DBF) ← 1

SEL RB0 Select Register Bank 0 (Not in 8021)

1 1 0 0	0 1 0 1

PSW bit 4 is set to zero. References to working registers 0-7 address data memory locations 0-7. This is the recommended setting for normal program execution.

(BS) ← 0

SEL RB1 Select Register Bank 1 (Not in 8021)

1 1 0 1	0 1 0 1

PSW bit 4 is set to one. References to working registers 0-7 address data memory locations 24-31. This is the recommended setting for interrupt service routines, since locations 0-7 are left intact. The setting of PSW bit 4 in effect at the time of an interrupt is restored by the RETR instruction when the interrupt service routine is completed.

(BS) ← 1

Example: Assume an external interrupt has occurred, control has passed to program memory location 3, and PSW bit 4 was zero before the interrupt.

```
LOC3: JNI INIT    ;JUMP TO ROUTINE 'INIT' IF
                  ;INTERRUPT INPUT IS ZERO
```

Mnemonics copyright Intel Corporation 1976.

```
INIT: MOV R7,A          ;MOVE ACC CONTENTS TO
                        ;LOCATION 7
      SEL RB1           ;SELECT REG BANK 1
      MOV R7,#0FAH      ;MOVE 'FA' HEX TO LOCATION 31
      .
      .
      .
      SEL RB0           ;SELECT REG BANK 0
      MOV A,R7          ;RESTORE ACC FROM LOCATION 7
      RETR              ;RETURN — RESTORE PC AND PSW
```

STOP TCNT Stop Timer/Event-Counter

0 1 1 0	0 1 0 1

This instruction is used to stop both time accumulation and event counting.

Example: Disable interrupt, but jump to interrupt routine after eight overflows and stop timer. Count overflows in register 7.

```
START:  DIS TCNTI       ;DISABLE TIMER INTERRUPT
        CLR A           ;CLEAR ACC TO ZEROS
        MOV T,A         ;MOVE ZEROS TO TIMER
        MOV R7,A        ;MOVE ZEROS TO REG 7
        STRT T          ;START TIMER
MAIN:   JTF COUNT       ;JUMP TO ROUTINE 'COUNT'
                        ;IF TF=1 AND CLEAR TIMER FLAG
        JMP MAIN        ;CLOSE LOOP
COUNT:  INC R7          ;INCREMENT REG 7
        MOV A,R7        ;MOVE REG 7 CONTENTS TO ACC
        JB3 INT         ;JUMP TO ROUTINE 'INT' IF ACC
                        ;BIT 3 IS SET (REG 7=8)
        JMP MAIN        ;OTHERWISE RETURN TO ROUTINE
                        ;MAIN
        .
        .
        .
INT:    STOP TCNT       ;STOP TIMER
        JMP 7H          ;JUMP TO LOCATION 7 (TIMER)
                        ;INTERRUPT ROUTINE
```

Mnemonics copyright Intel Corporation 1976.

STRT CNT Start Event Counter

0 1 0 0	0 1 0 1

The test 1 (T1) pin is enabled as the event-counter input and the counter is started. The event-counter register is incremented with each high-to-low transition on the T1 pin.

Example: Initialize and start event counter. Assume overflow is desired with first T1 input.

```
STARTC: EN TCNTI      ;ENABLE COUNTER INTERRUPT
        MOV A,#0FFH   ;MOVE 'FF' HEX (ONES) TO
                      ;ACC
        MOV T,A       ;MOVE ONES TO COUNTER
        STRT CNT      ;ENABLE T1 AS COUNTER
                      ;INPUT AND START
```

STRT T Start Timer

0 1 0 1	0 1 0 1

Timer accumulation is initiated in the timer register. The register is incremented every 32 instruction cycles. The prescaler which counts the 32 cycles is cleared but the timer register is not.

Example: Initialize and start timer.

```
STARTT: CLR A         ;CLEAR ACC TO ZEROS
        MOV T,A       ;MOVE ZEROS TO TIMER
        EN TCNTI      ;ENABLE TIMER INTERRUPT
        STRT T        ;START TIMER
```

SWAP A Swap Nibbles Within Accumulator

0 1 0 0	0 1 1 1

Bits 0-3 of the accumulator are swapped with bits 4-7 of the accumulator.

$(A_{4-7}) \rightleftarrows (A_{0-3})$

Example: Pack bits 0-3 of locations 50-51 into location 50.

```
PCKDIG: MOV R0, #50   ;MOVE '50' DEC TO REG 0
        MOV R1, #51   ;MOVE '51' DEC TO REG 1
        XCHD A,@R0    ;EXCHANGE BITS 0-3 OF ACC
                      ;AND LOCATION 50
        SWAP A        ;SWAP BITS 0-3 AND 4-7 OF ACC
        XCHD A,@R1    ;EXCHANGE BITS 0-3 OF ACC AND
                      ;LOCATION 51
        MOV @R0,A     ;MOVE CONTENTS OF ACC TO
                      ;LOCATION 50
```

Mnemonics copyright Intel Corporation 1976.

The contents of the accumulator and the contents of working register 'r' are exchanged.

(A) ⇄ (Rr) r=0-7

Example: Move PSW contents to Reg 7 without losing accumulator contents.

```
XCHAR7: XCH A,R7     ;EXCHANGE CONTENTS OF REG 7
                     ;AND ACC
        MOV A, PSW   ;MOVE PSW CONTENTS TO ACC
        XCH A,R7     ;EXCHANGE CONTENTS OF REG 7
                     ;AND ACC AGAIN
```

XCH A,@R_r Exchange Accumulator and Data Memory Contents

0 0 1 0	0 0 0 r

The contents of the accumulator and the contents of the resident data memory location addressed by bits 0-5*of register 'r' are exchanged. Register 'r' contents are unaffected.

(A) ⇄ — ((Rr)) r=0-1

Example: Decrement contents of location 52.

```
DEC52: MOV R0,#52    ;MOVE '52' DEC TO ADDRESS
                     ;REG 0
       XCH A,@R0     ;EXCHANGE CONTENTS OF ACC
                     ;AND LOCATION 52
       DEC A         ;DECREMENT ACC CONTENTS
       XCH A,@R0     ;EXCHANGE CONTENTS OF ACC
                     ;AND LOCATION 52 AGAIN
```

XCHD A,@R_r Exchange Accumulator and Data Memory 4-Bit Data

0 0 1 1	0 0 0 r

This instruction exchanges bits 0-3 of the accumulator with bits 0-3 of the data memory location addressed by bits 0-5*of register 'r'. Bits 4-7 of the accumulator, bits 4-7 of the data memory location, and the contents of register 'r' are unaffected.

(A_{0-3}) ⇄ ((Rr0-3)) r=0-1

Mnemonics copyright Intel Corporation 1976.

*0-6 for 8039/8049

Example: Assume program counter contents have been stacked in locations 22-23.

```
XCHNIB: MOV R0,#23      ;MOVE '23' DEC TO REG 0
        CLR A           ;CLEAR ACC TO ZEROS
        XCHD A,@R0      ;EXCHANGE BITS 0-3 OF ACC
                        ;AND LOCATION 23 (BITS 8-11
                        ;OF PC ARE ZEROED, ADDRESS
                        ;REFERS TO PAGE 0)
```

XRL A,R_r Logical XOR Accumulator With Register Mask

1 1 0 1	1 r r r

Data in the accumulator in EXCLUSIVE ORed with the mask contained in working register 'r'.

(A) ← (A) XOR (Rr) r=0-7

```
Example: XORREG: XRL A,R5      ;'XOR' ACC CONTENTS WITH
                               ;MASK IN REG 5
```

XRL A,@R_r Logical XOR Accumulator With Memory Mask

1 1 0 1	0 0 0 r

Data in the accumulator is EXCLUSIVE ORed with the mask contained in the data memory location addressed by register 'r', bits 0-5*.

(A) ← (A) XOR ((Rr)) r=0-1

```
Example: XORDM: MOV R1, #20H   ;MOVE '20' HEX TO REG 1
                XRL A,@R1      ;'XOR' ACC CONTENTS WITH MASK
                               ;IN LOCATION 32
```

XRL A,#data Logical XOR Accumulator With Immediate Mask

1 1 0 1	0 0 1 1	d_7 d_6 d_5 d_4	d_3 d_2 d_1 d_0

This is a 2-cycle instruction. Data in the accumulator is EXCLUSIVE ORed with an immediately-specified mask.

(A) ← (A) XOR data

```
Example: XORID: XOR A,#HEXTEN  ;XOR CONTENTS OF ACC WITH
                               ;MASK EQUAL VALUE OF SYMBOL
                               ;'HEXTEN'
```

Mnemonics copyright Intel Corporation 1976.

*0-6 for 8039/8049

Chapter 5
APPLICATION EXAMPLES



APPLICATION EXAMPLES

5.0 Introduction 5-1

5.1 Hardware Examples 5-1

- Frequency Reference Options 5-1
- Stand Alone 8048/8049 5-2
- Stand Alone 8021 5-3
- Multiple Interrupt Sources 5-4
- Multiple Prioritized Interrupts 5-5
- External Program Memory 5-6
- Program Memory and I/O Expansion 5-7
- Data Memory and I/O Expansion 5-8
- A Three Chip System 5-9
- I/O Expansion Techniques 5-10
- 8049 Emulator Circuit 5-16
- General Application Examples 5-18

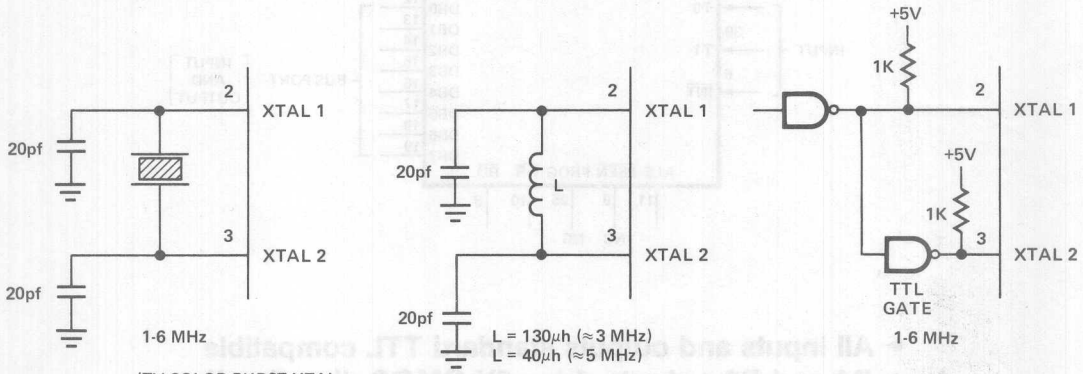
5.2 Software Examples 5-21

APPLICATION EXAMPLES

5.0 Introduction

The following chapter is organized in two sections, Hardware and Software. The hardware section gives examples of some typical configurations of MCS-48 components while software section gives assembly language listings of some common applications routines.

5.1 Hardware Examples

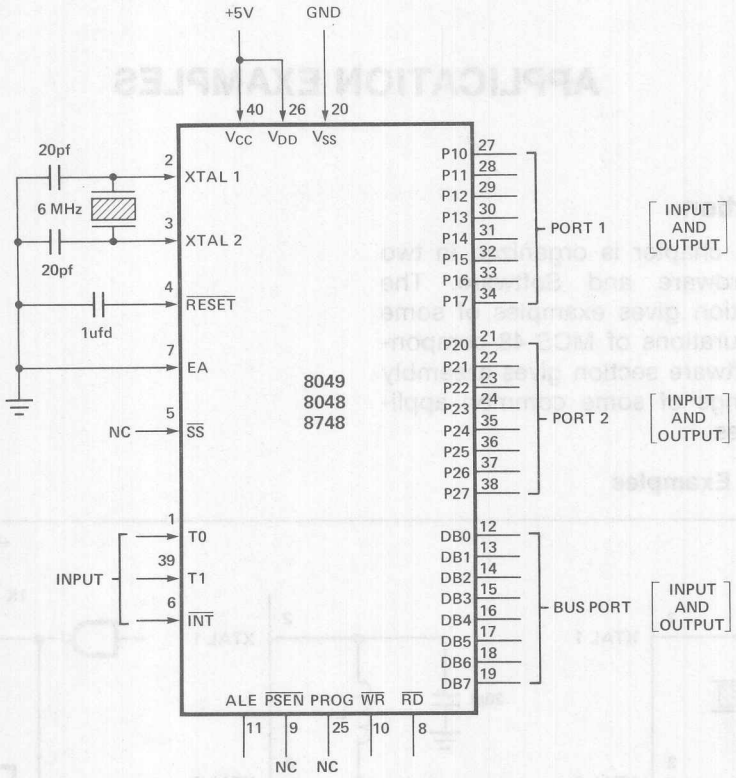


NOTE:
A STANDARD SERIES RESONANT XTAL SUCH AS CTS KNIGHTS MP060 OR CRYSTEK CY6B WILL PROVIDE BETTER THAN .1% FREQUENCY ACCURACY.

IF HIGHER ACCURACY IS REQUIRED A PARALLEL RESONANT XTAL SHOULD BE SPECIFIED WHICH WILL OPERATE WITH A LOAD CAPACITANCE OF 20-25pf. FOR MORE INFORMATION ON XTALS SEE INTEL APPLICATION NOTE AP-35.

NOTE: SEE PAGE 5-3 FOR 8021 FREQUENCY REFERENCES.

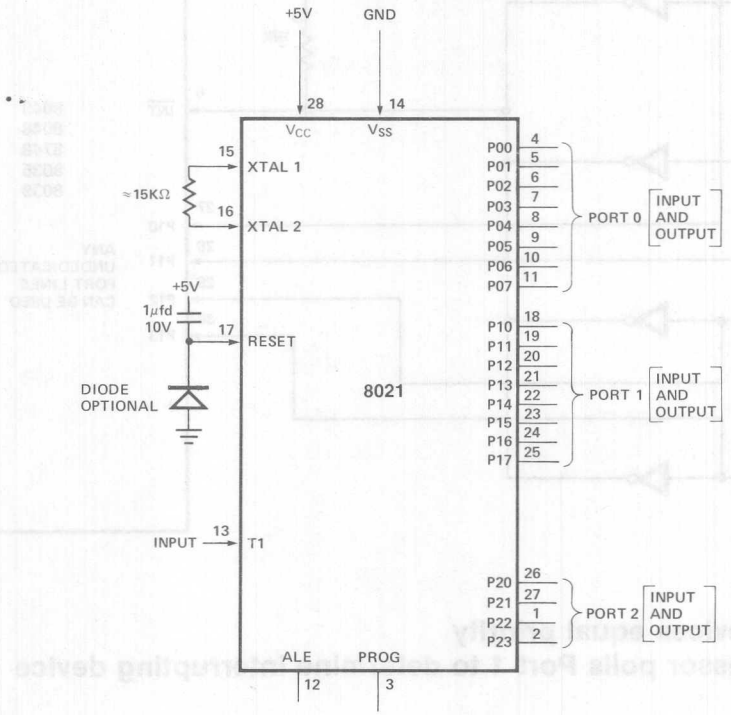
FREQUENCY REFERENCE OPTIONS

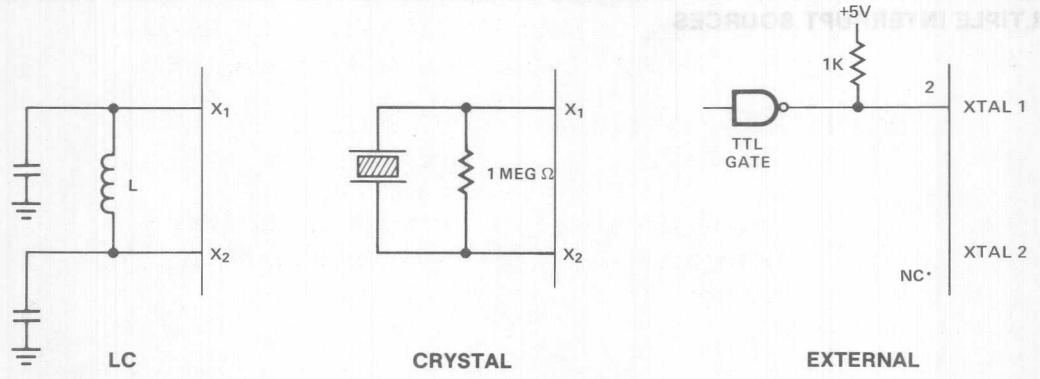


- **All inputs and outputs standard TTL compatible**
- **P1 and P2 outputs drive 5V CMOS directly others require 10 to 50KΩ pullup for CMOS compatibility**

XTAL: Series Resonant
1 to 6 MHz
or
Parallel Resonant
for higher
accuracy

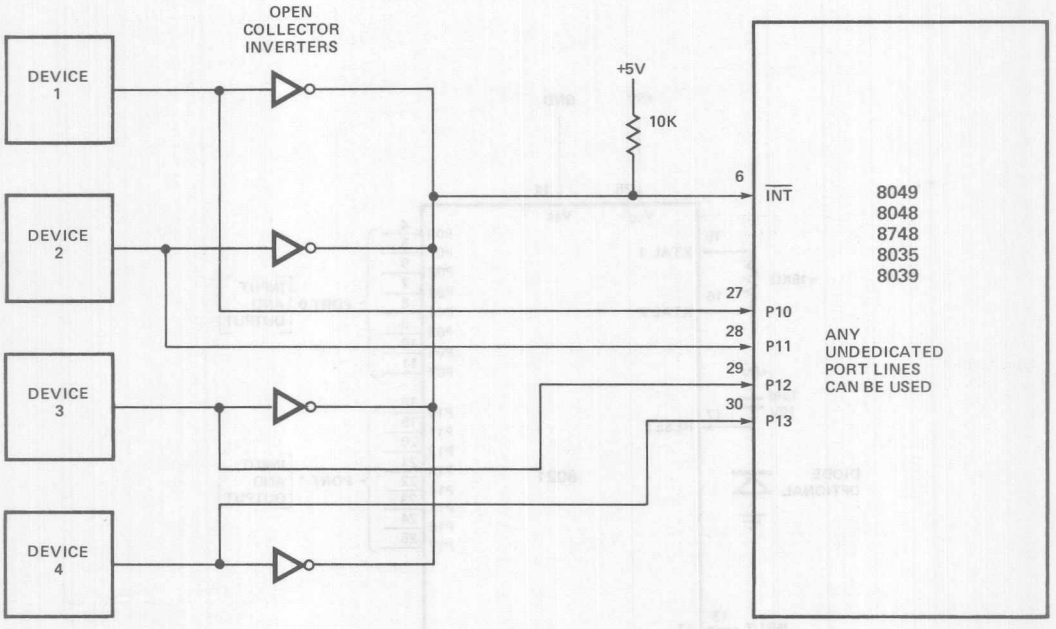
CTS Knights MP060
Crystek CY6B
or equivalent
or standard 3.58 MHz
TV Color Burst XTAL





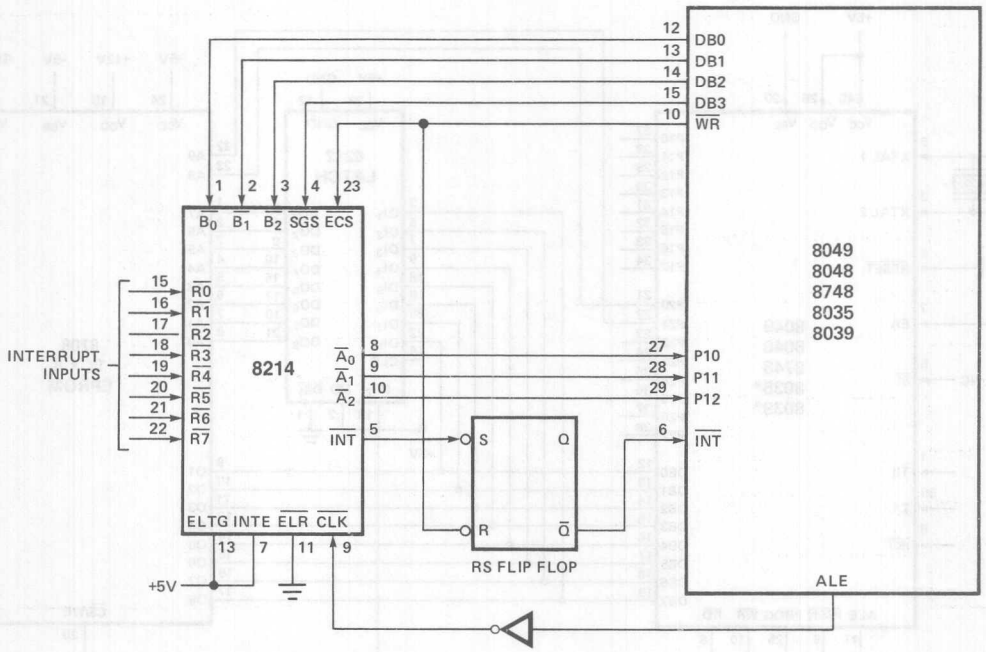
ALTERNATE FREQUENCY REFERENCE OPTIONS
(COMPONENT VALUES TO BE DETERMINED)

THE STAND ALONE 8021



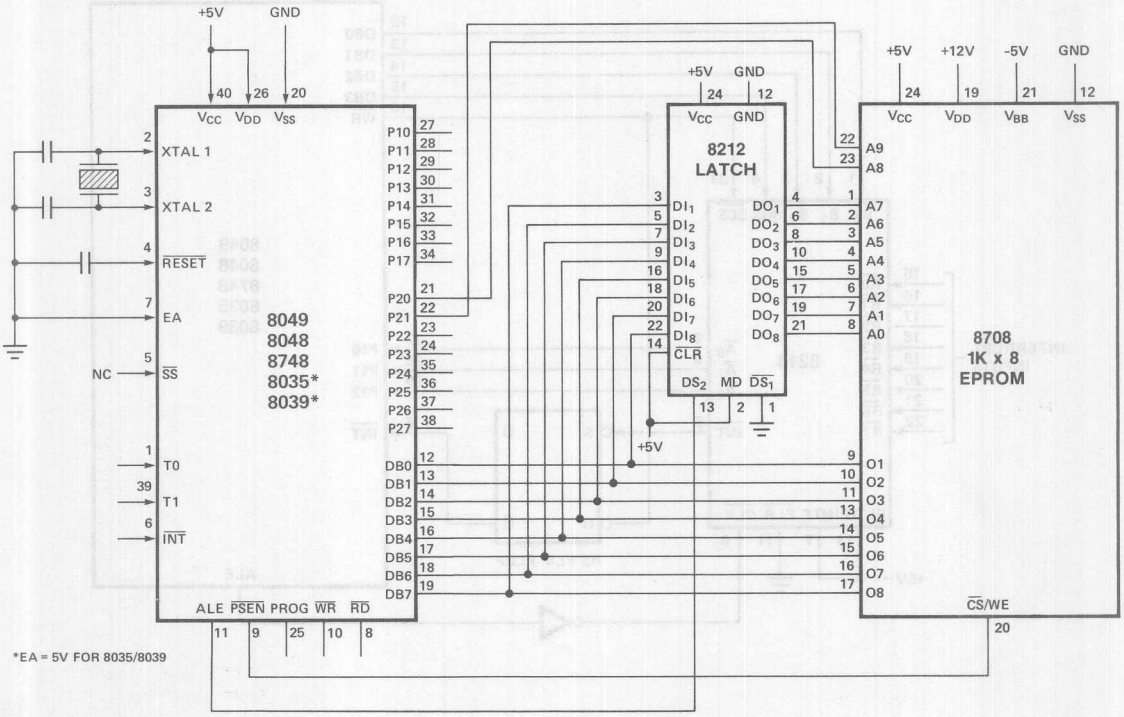
- **All devices equal priority**
- **Processor polls Port 1 to determine interrupting device**

MULTIPLE INTERRUPT SOURCES



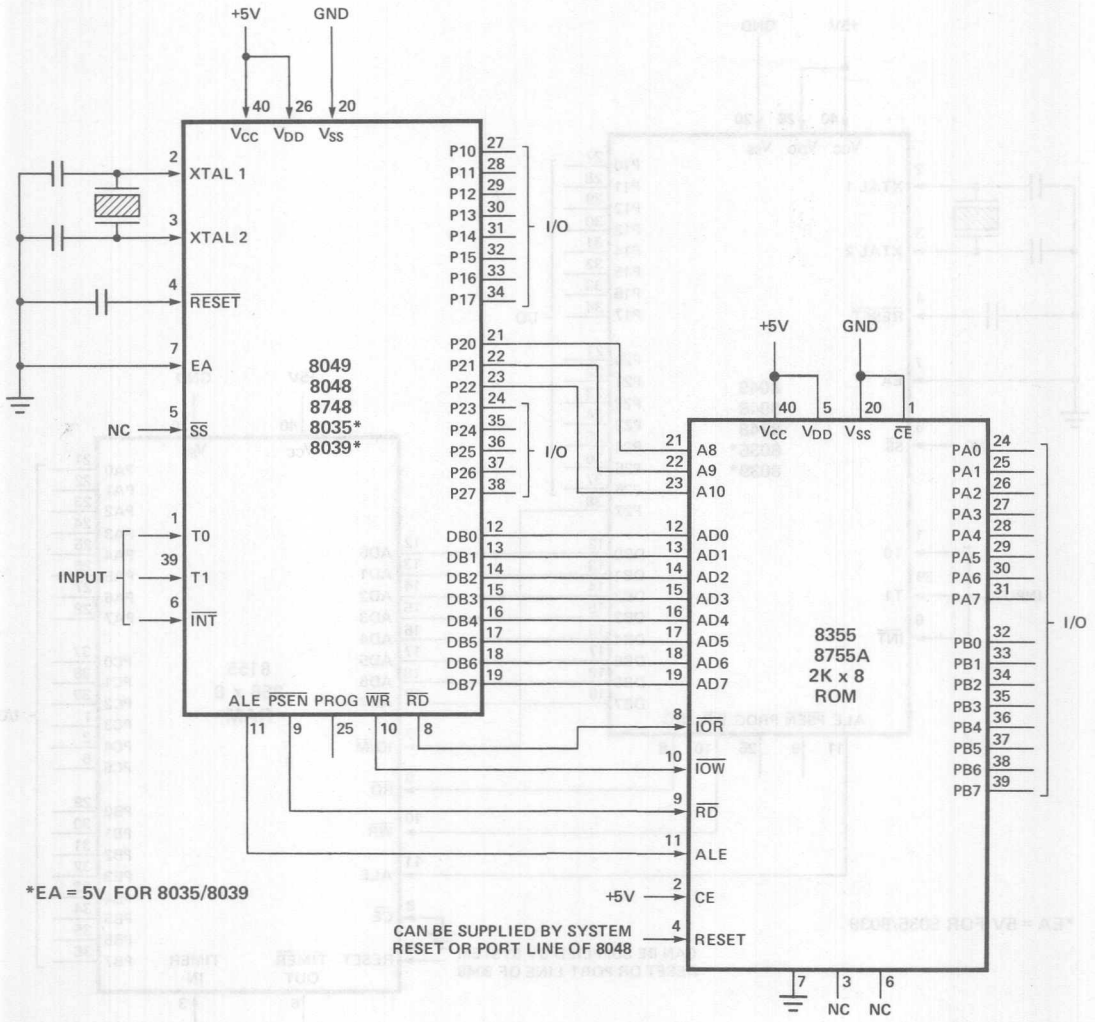
- **Processor polls Port 1 to determine interrupting device**
- **Processor sets priority level by writing 4-bits to 8214**

MULTIPLE INTERRUPTS WITH PRIORITY LEVELS



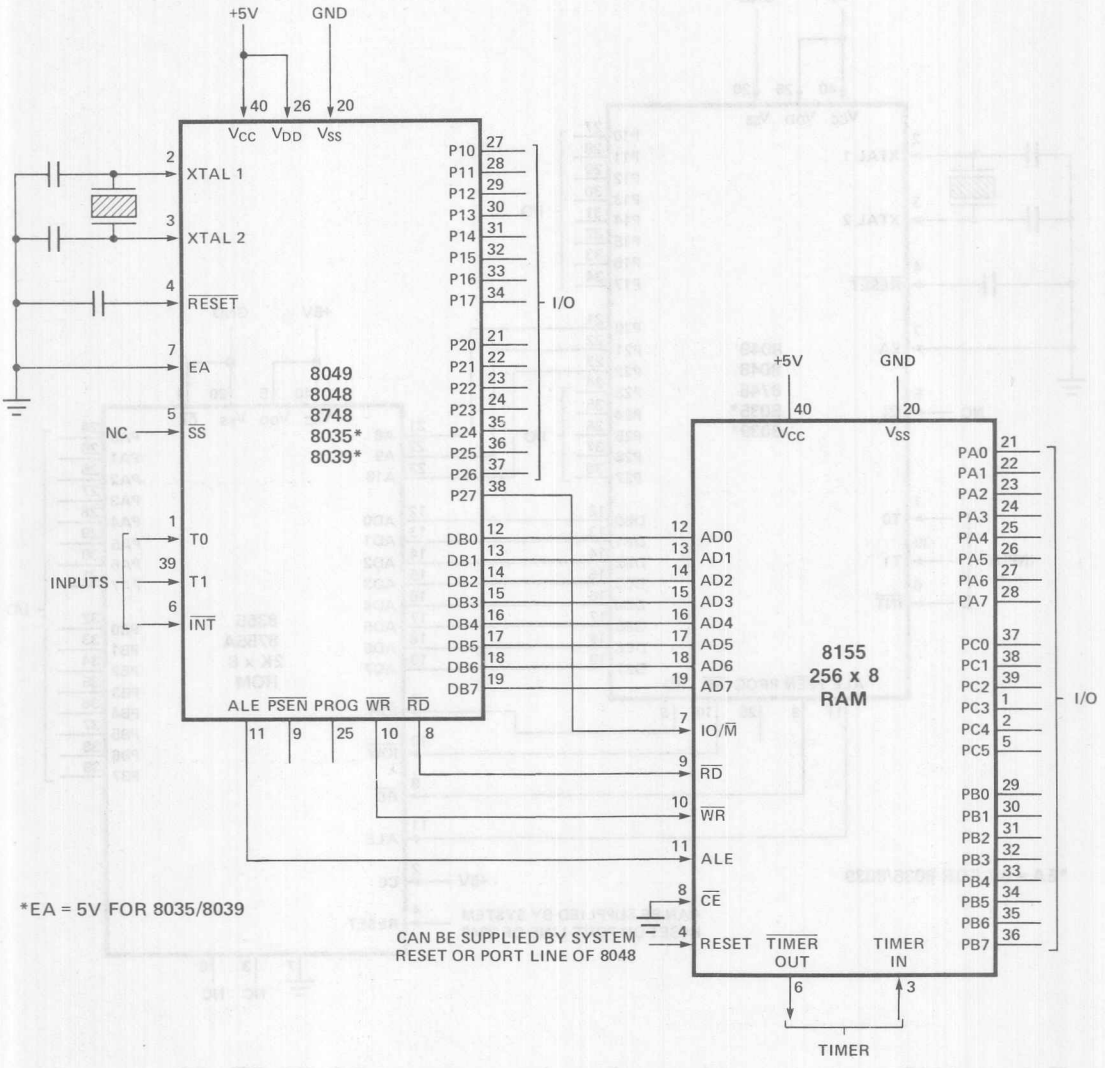
- 8212 serves as address latch
- Address is valid while ALE is high and is latched when ALE goes low

EXTERNAL PROGRAM MEMORY



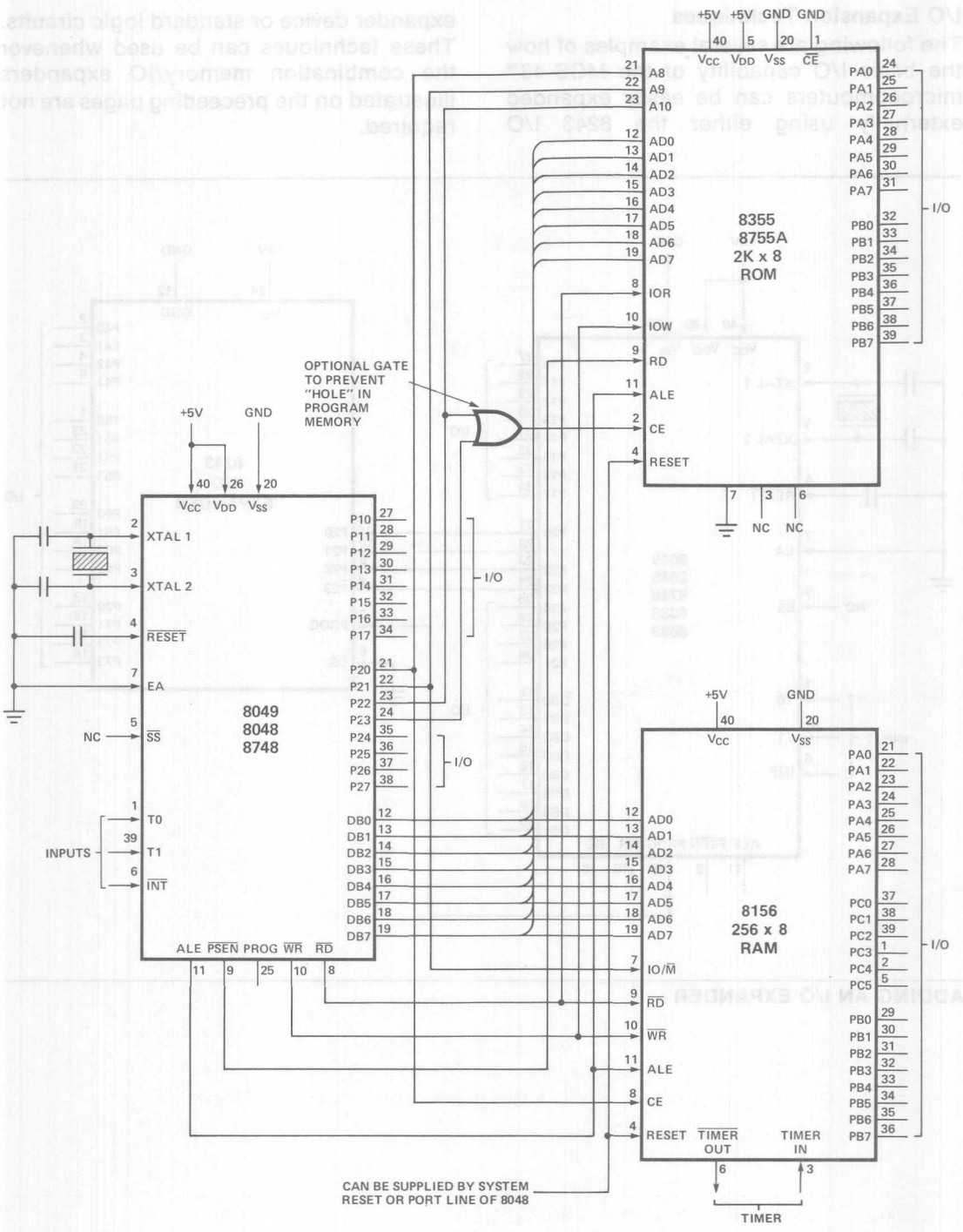
- **External I/O parts are addressed as data memory PA=00 PB=01**
- **If the 8048's internal Program Memory is used this configuration will result in the upper 1K of external memory being addressed before the lower 1K. Inverting A10 will correct this if necessary. This inversion is not necessary if the 8049 is used.**

ADDING A PROGRAM MEMORY AND I/O EXPANDER



- **Both I/O and RAM are addressed as data memory**
- **Writing a bit to P27 determines whether RAM or I/O is to be accessed**

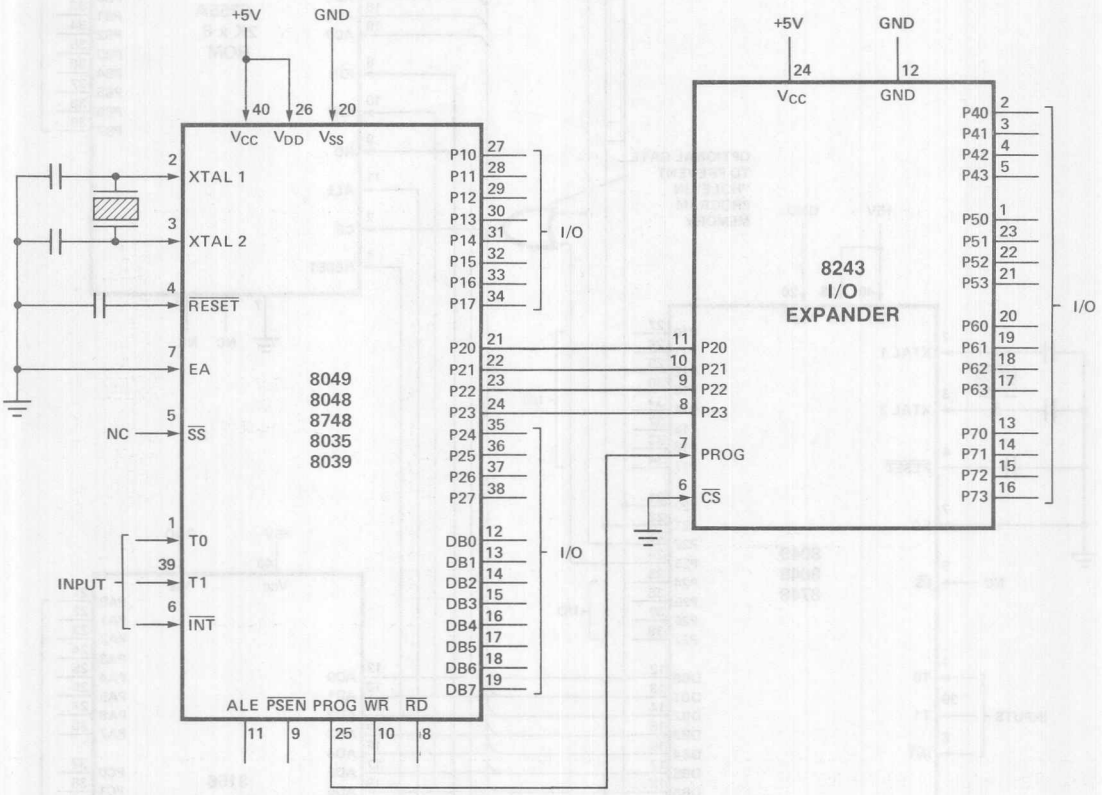
ADDING A DATA MEMORY AND I/O EXPANDER



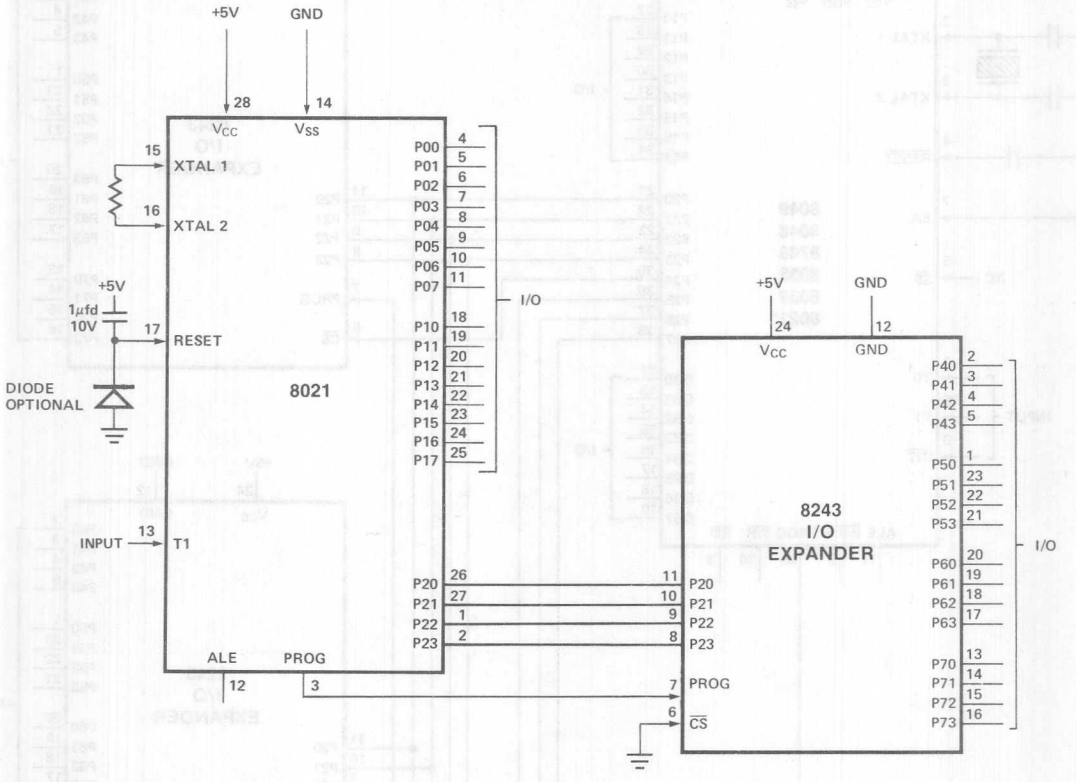
- **This configuration is explained in section 3.4**

I/O Expansion Techniques

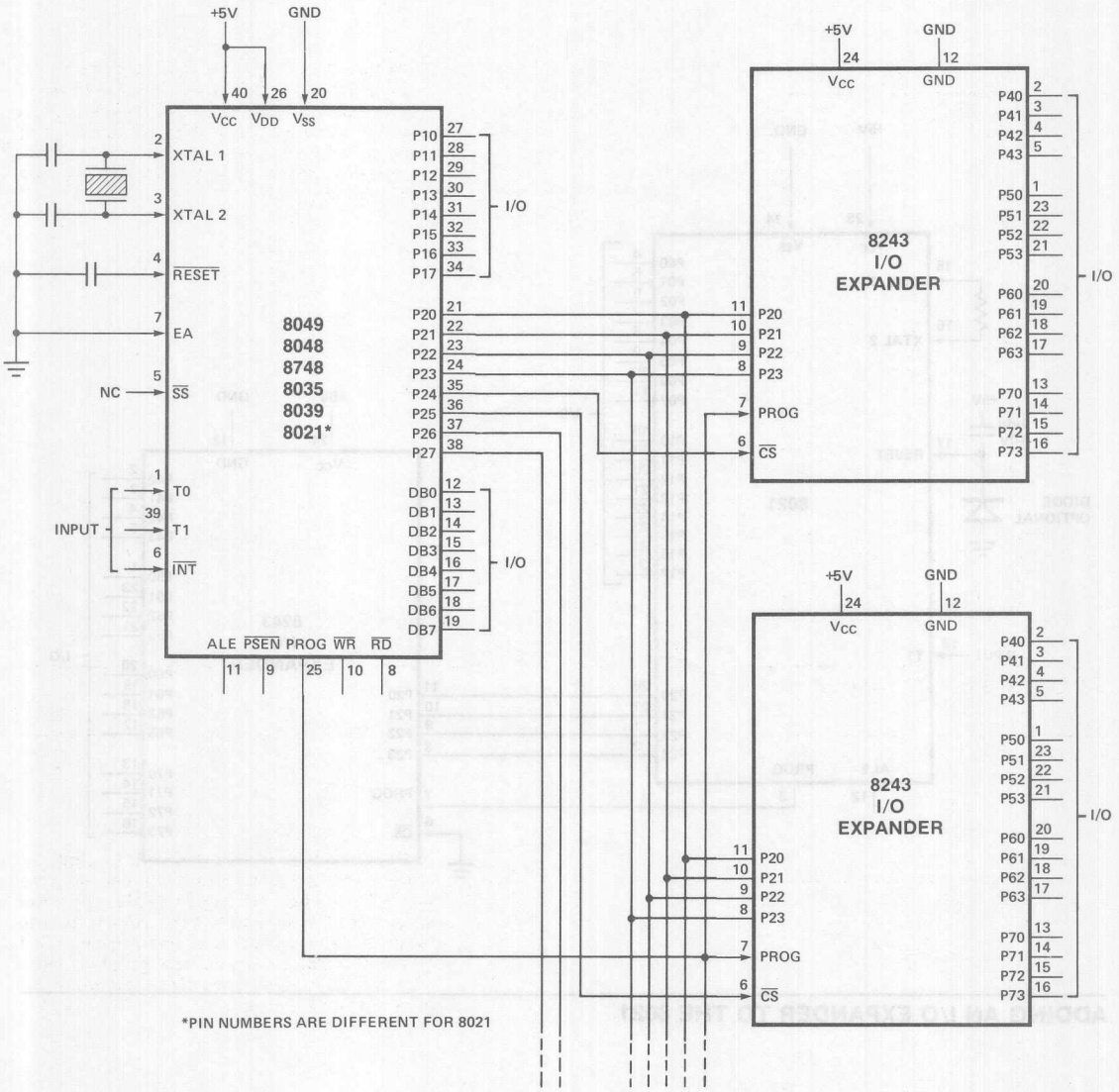
The following are several examples of how the basic I/O capability of the MCS-48™ microcomputers can be easily expanded externally using either the 8243 I/O expander device or standard logic circuits. These techniques can be used whenever the combination memory/IO expanders illustrated on the preceeding pages are not required.



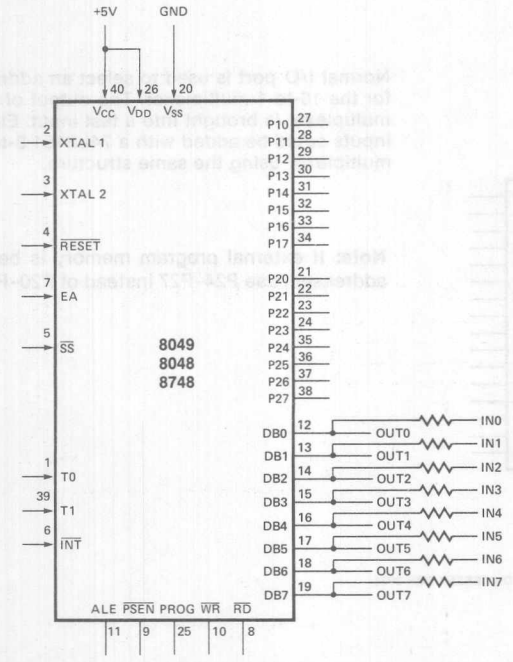
ADDING AN I/O EXPANDER



ADDING AN I/O EXPANDER TO THE 8021

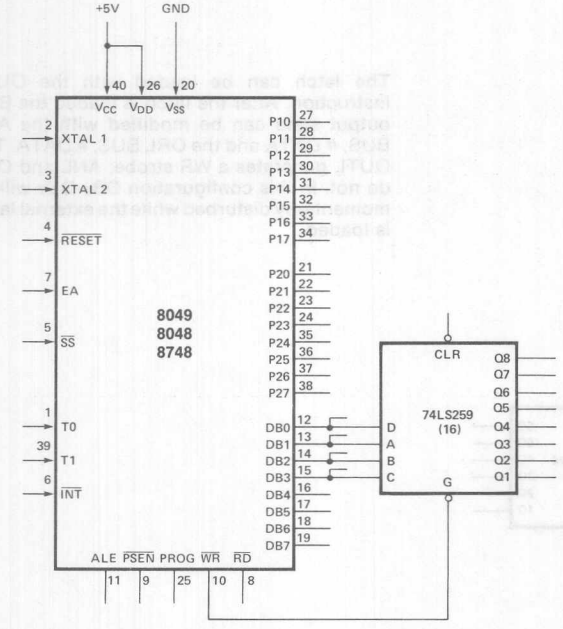


ADDING MULTIPLE I/O EXPANDERS



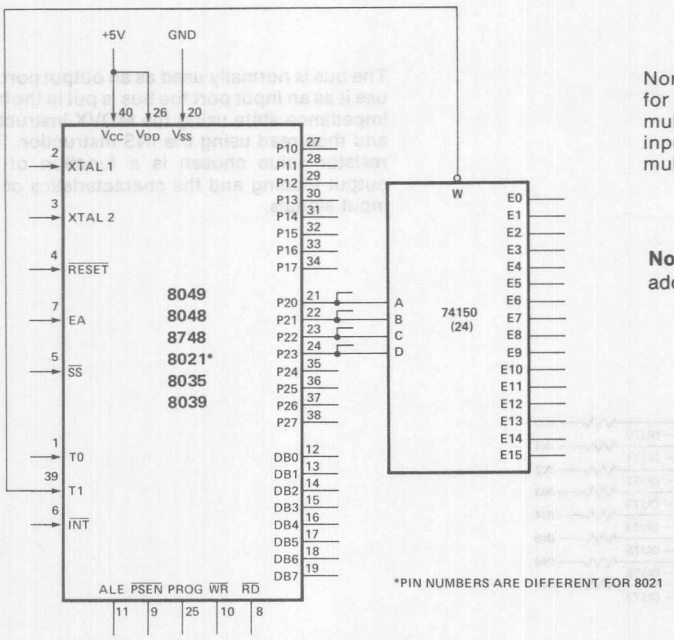
The bus is normally used as an output port. To use it as an input port the bus is put in the high impedance state using the MOVX instruction and then read using the INS instruction. The resistor value chosen is a function of the output loading and the characteristics of the input signals.

ADDING 8 INPUT LINES



Individual bits of the 74LS259 eight-bit addressable latch can be set or reset using the OUTL instruction. During the OUTL operation bit zero of the accumulator is written into the bit of the latch specified by bits 1 through 3 of the accumulator. In this configuration DB_0-DB_7 will be momentarily disturbed while the external latch is loaded.

ADDING 8 OUTPUT LINES

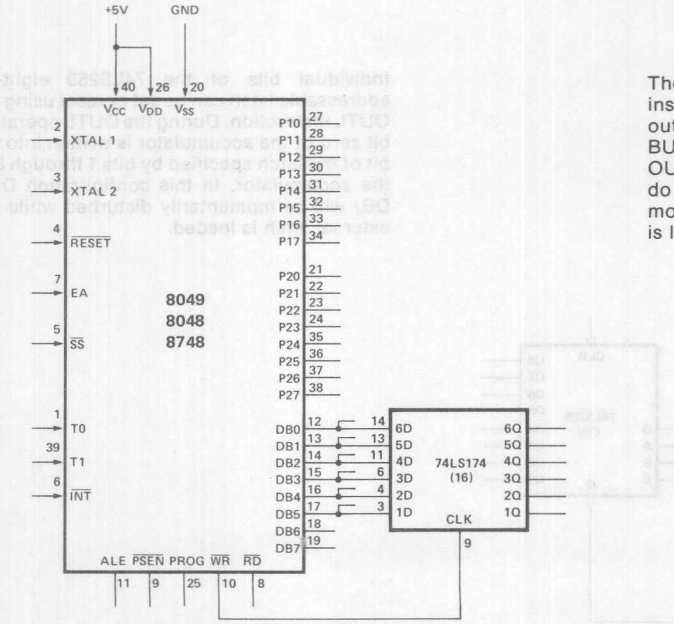


*PIN NUMBERS ARE DIFFERENT FOR 8021

Normal I/O port is used to select an address for the 16-to-1 multiplexer. The output of the multiplexer is brought into a test input. Eight inputs could be added with a 74LS151 8-to-1 multiplexer using the same structure.

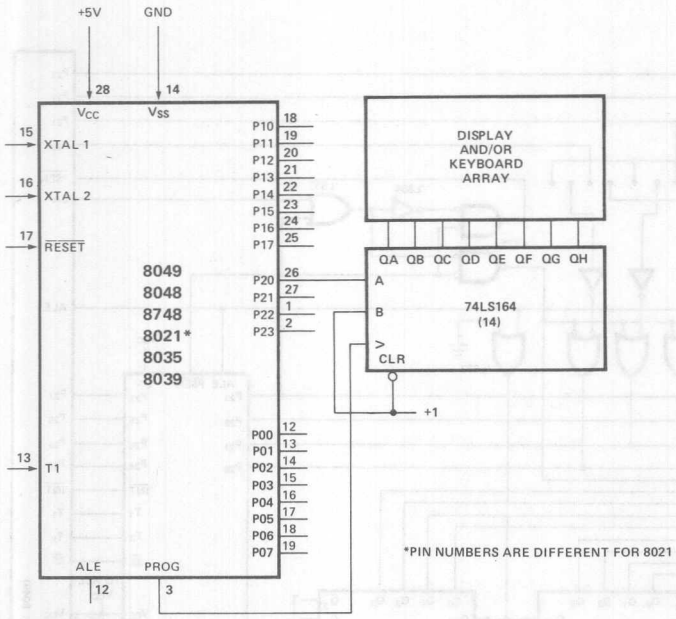
Note: If external program memory is being addressed, use P24–P27 instead of P20–P23.

ADDING 16 INPUT LINES



The latch can be loaded with the OUTL instruction. After the latch is loaded the BUS output state can be modified with the ANL BUS, # DATA and the ORL BUS, # DATA. The OUTL generates a WR strobe; ANL and ORL do not. In this configuration DB_0-DB_7 will be momentarily disturbed while the external latch is loaded.

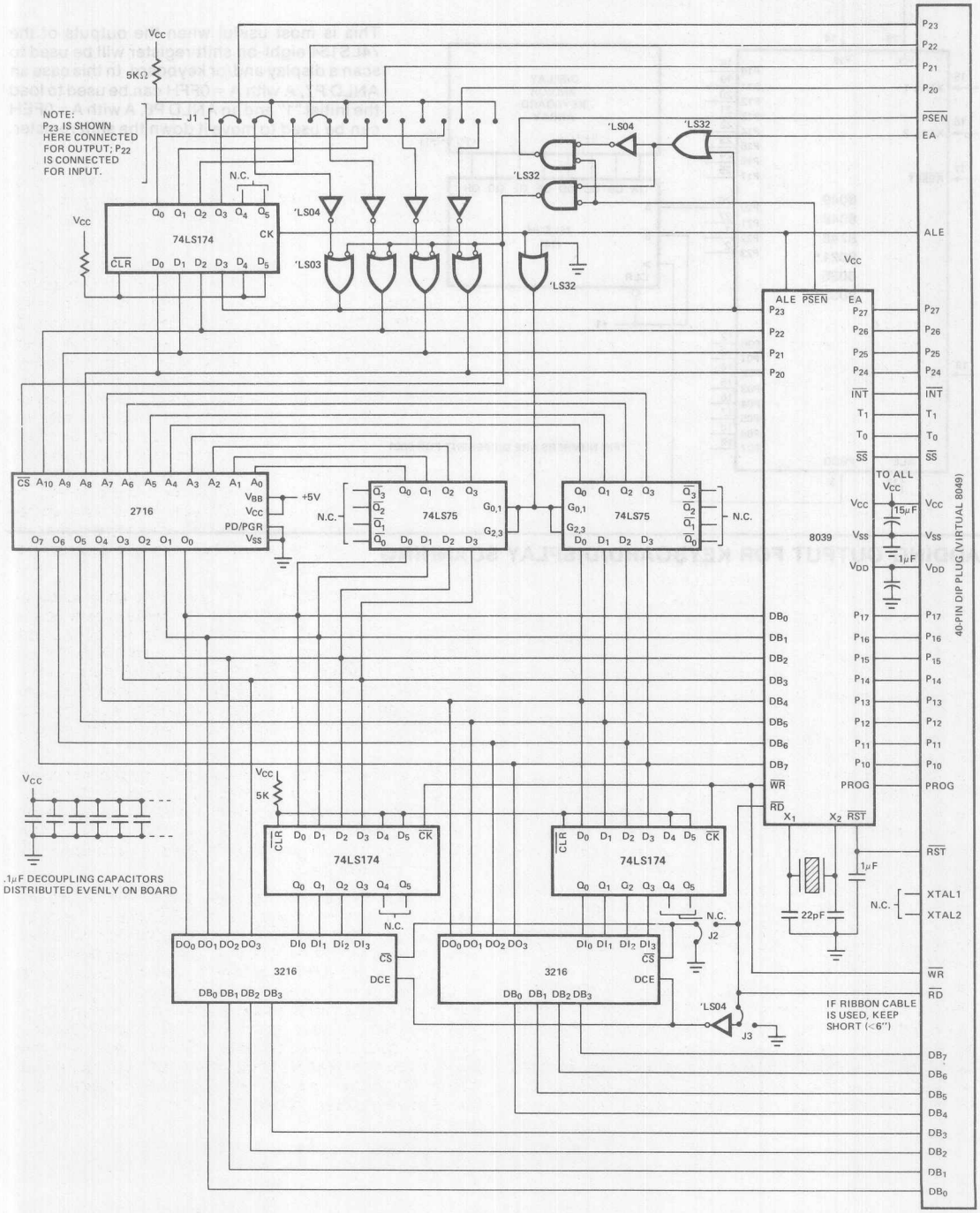
ADDING 6 OUTPUT LINES



*PIN NUMBERS ARE DIFFERENT FOR 8021

This is most useful when the outputs of the 74LS164 eight-bit shift register will be used to scan a display and/or keyboard. In this case an ANLD P7, A with A = 0FFH can be used to load the initial "1" and an ANLD P6, A with A = 0FEH can be used to move it down the shift register.

ADDING OUTPUT FOR KEYBOARD/DISPLAY SCANNING



8049 EMULATOR CIRCUIT

8049 EMULATOR CIRCUIT DESCRIPTION

The following is an explanation of a circuit which emulates the operation of an Intel® 8049 using a standard EPROM for program storage.

With the 8049, software may be developed by running external program memory, but doing so requires the use of the bus and P_{23}-P_{20} to access this memory.

The circuit shown may be used to restore the normal functioning of these twelve I/O pins. The circuit consists of an 8039 CPU, 2716 EPROM, two 8216 bi-directional bus drivers, and eight other 7400 Series Low-Power Schottky TTL packages. The whole assembly can be built on a 2-3/4" x 4" board.

A cable coming off the board can be terminated by a forty-pin plug which may be inserted directly into the CPU socket intended for the 8049 in a system undergoing design or testing. Alternatively, a pattern of forty pins extending below the board can be used to plug the board directly into the system undergoing testing, "piggy-back" fashion. The emulator board may be configured in various ways so that the 40 pin plug is the logical equivalent of an 8049 in every legal operating mode. (In the following explanation of the operation of the circuit, an asterisk appearing before a signal or pin number — as in *$\overline{PSEN}$ — refers to that pin on the "virtual 8049" represented by the forty-pin plug).

Since the CPU is identical with the 8049 in all respects other than its lack of program memory, most of the pins of the 8039 are simply connected directly to the corresponding pins of the forty-pin plug. These include all of Port 1, the high order bits of Port 2, the test pins, etc. Signals which are emulated with additional logic include the rest of Port 2, DB_7-DB_0, *$\overline{PSEN}$, etc. $\overline{RD}$, $\overline{WR}$, ALE, and $\overline{PSEN}$ are obtained from the 8039, but are also used by the emulation circuitry.

The EA input of the 8039 is hard-wired high so all instruction fetches are made from the 2716. Two 74LS75 four-bit latches gated by the buffered ALE signal are used to hold the lower eight bits of address from the time-multiplexed data bus. Since the Bus is being used for fetching instructions, data latched to the Bus will be lost on the next instruction fetch. Two 74LS174 latches are used to retain the output data when a bus write is executed. These latches are triggered by the trailing edge of the $\overline{WR}$ pulse, so their outputs are glitch free. Since logical operations to the bus do not generate a WR strobe, the "ANL BUS,#" and "ORL BUS,#" instructions may not be used, though they do function properly with the other ports.

The two 8216 bi-directional bus drivers normally buffer the latched bus contents to the DB pins of the virtual 8049. When an "INS A,BUS" instruction is executed, they buffer the input signals on to the emulator data bus. Thus, the circuit is designed to use the Bus for both latched output and strobed input. If $DB_7$9DB_0 of the 8049 are to be used solely for input data, J2 and J3 may be changed from what is shown in the Figure, so that DB_7-DB_0 act as high impedance inputs and the 8216s are enabled only when the read operation is performed. If the bus is to be used only for latched output, the 8216s can be omitted entirely.

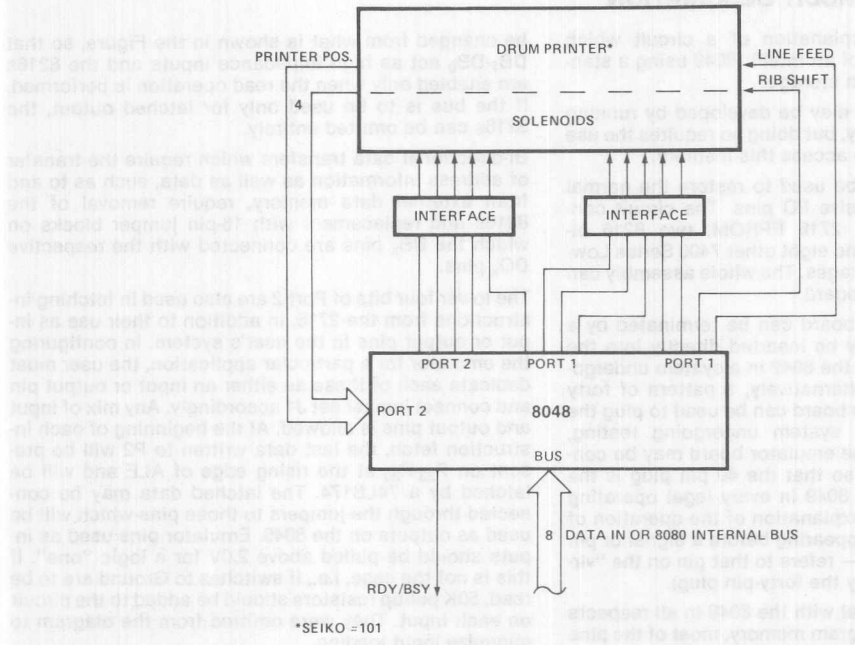
Bi-directional data transfers which require the transfer of address information as well as data, such as to and from external data memory, require removal of the 8216s and replacement with 16-pin jumper blocks on which the DB_x pins are connected with the respective DO_x pins.

The lower four bits of Port 2 are also used in fetching instructions from the 2716, in addition to their use as input or output pins in the user's system. In configuring the emulator for a particular application, the user must dedicate each of these as either an input or output pin and connect jumper set J1 accordingly. Any mix of input and output pins is allowed. At the beginning of each instruction fetch, the last data written to P2 will be present on P_{23}-P_{20} at the rising edge of ALE and will be latched by a 74LS174. The latched data may be connected through the jumpers to those pins which will be used as outputs on the 8049. Emulator pins used as inputs should be pulled above 2.0V for a logic "one". If this is not the case, i.e., if switches to Ground are to be read, 50K pullup resistors should be added to the circuit on each input. They were omitted from the diagram to minimize input loading.

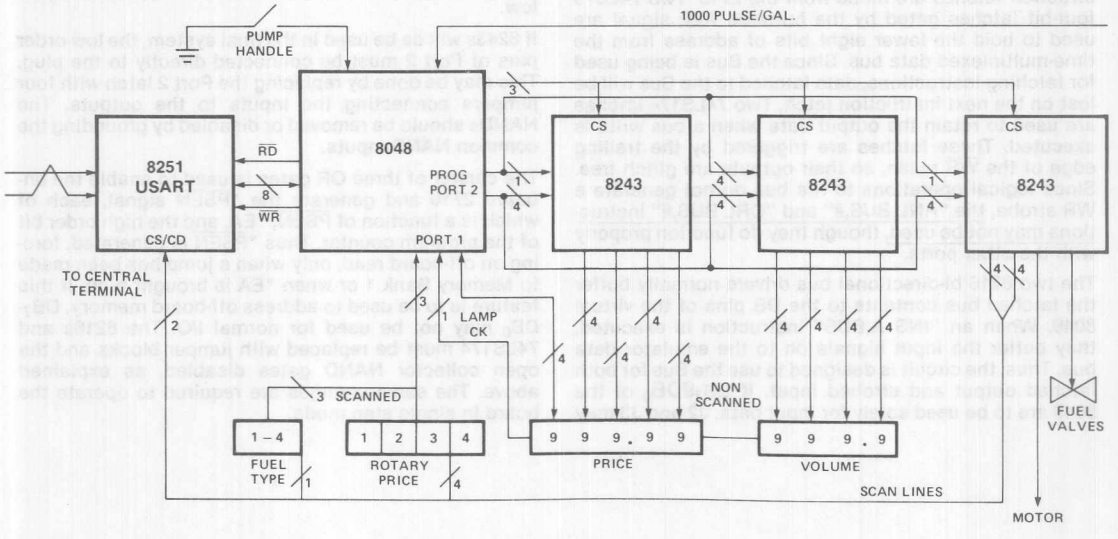
Pins which will be used as inputs may be connected to the input of an OR gate formed of inverters and open-collector NAND gates. The input signals will be relayed directly to the 8039 and will be read by an "IN A,P2" instruction. But when $\overline{PSEN}$ is low, the NAND outputs are forced off, allowing the 8039 pins to be used for high-order program adressing. Open-collector outputs are needed to prevent line contention when $\overline{PSEN}$ is not low.

If 8243s will be be used in the final system, the low order pins of Port 2 must be connected directly to the plug. This may be done by replacing the Port 2 latch with four jumpers connecting the inputs to the outputs. The NANDs should be removed or disabled by grounding the common NAND inputs.

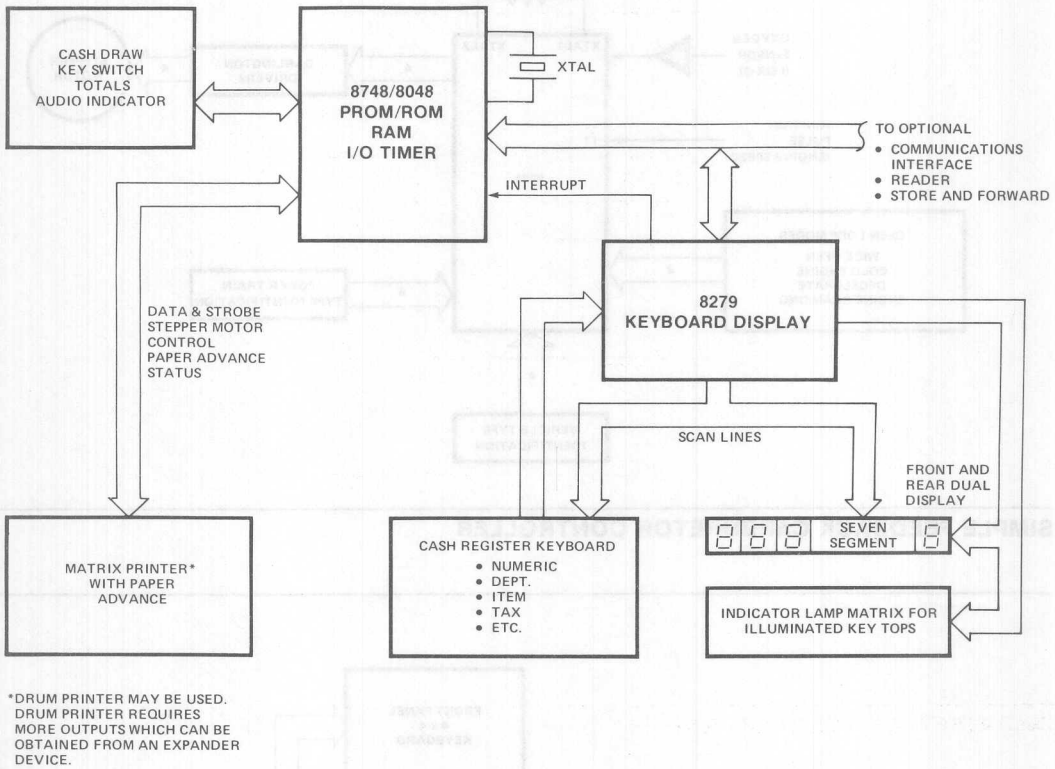
The cluster of three OR gates is used to enable the on-board 2716 and generate the *$\overline{PSEN}$ signal, each of which is a function of $\overline{PSEN}$, *EA, and the high order bit of the program counter. Thus *$\overline{PSEN}$ is generated, forcing an off-board read, only when a jump has been made to Memory Bank 1 or when *EA is brought high. If this feature is to be used to address off-board memory, DB_7-DB_0 may not be used for normal I/O. The 8216s and 74LS174 must be replaced with jumper blocks and the open collector NAND gates disabled, as explained above. The same changes are required to operate the board in single step mode.



8048 INTERFACE TO DRUM PRINTER

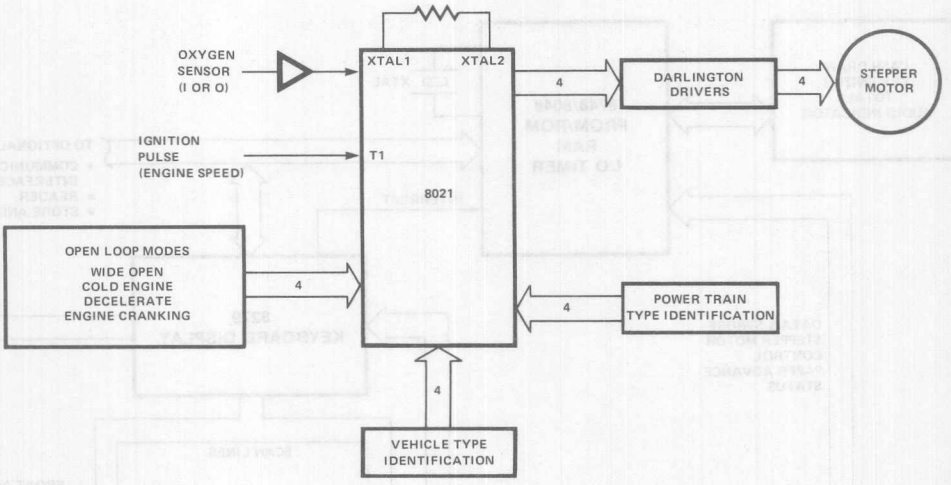


MCS-48™ GAS PUMP

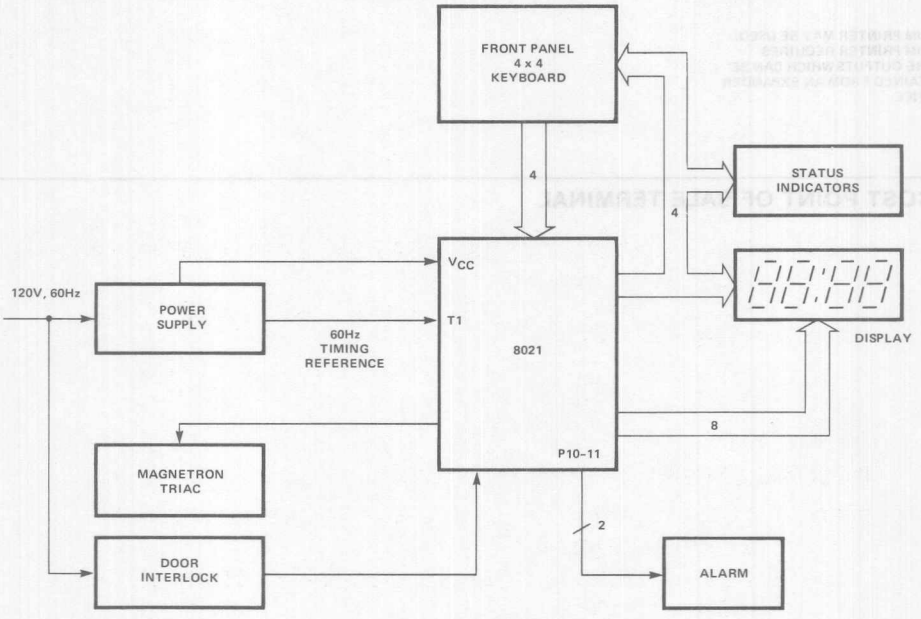


*DRUM PRINTER MAY BE USED. DRUM PRINTER REQUIRES MORE OUTPUTS WHICH CAN BE OBTAINED FROM AN EXPANDER DEVICE.

LOW COST POINT OF SALE TERMINAL



SIMPLE FEEDBACK CARBURETOR CONTROLLER



MICROWAVE OVEN CONTROLLER

5.2 Software Examples

The following routines are written as subroutines. R0 and R1 are used as data pointers, R2 is used as an extension of the accumulator and R3 is used as a loop counter.

RX0 = R0
AEX = R2

DOUBLE ADD

```
DADD:   DEC     RX0         ;GET LOW BYTE AND ADD TO A
        ADD     A,@RX0
        INC     RX0         ;GET HI BYTE AND ADD TO AEX
        XCH     A,AEX
        ADDC    A,@RX0
        XCH     A,AEX
        RET                 ;RETURN
```

DOUBLE SUBTRACT

```
DMIN:   DEC     RX0         ;GET LOW BYTE AND SUB FROM A
        CPL     A
        ADD     A,@RX0
        CPL     A
        INC     RX0         ;GET HI BYTE AND SUB FROM AEX
        XCH     A,AEX
        CPL     A
        ADDC    A,@RX0
        CPL     A
        XCH     A,AEX
        RET                 ;RETURN
```

DOUBLE LOAD

```
DLD:    DEC     RX0         ;GET LOW BYTE AND PLACE IN A
        MOV     A,@RX0
        INC     RX0         ;GET HI BYTE AND PLACE IN AEX
        XCH     A,AEX
        MOV     A,@RX0
        XCH     A,AEX
        RET                 ;RETURN
```

DOUBLE STORE

```
DST:    DEC     RX0         ;MOVE A INTO LOW BYTE
        MOV     @RX0,A
        INC     RX0         ;MOVE AEX INTO HIGH BYTE
        XCH     A,AEX
        MOV     @RX0,A
        XCH     A,AEX
        RET                 ;RETURN
```

DOUBLE EXCHANGE

```
DEX:    DEC     RX0         ;EXCHANGE A AND LOW BYTE
        XCH     A,@RX0
        INC     RX0         ;EXCHANGE AEX AND HIGH BYTE
        XCH     A,AEX
        XCH     A,@RX0
        XCH     A,AEX
        RET                 ;RETURN
```

DOUBLE LEFT LOGICAL SHIFT

```
LLSH:   RLC     A           ;SHIFT A
        XCH     A,AEX       ;SHIFT AEX
        RLC     A
        XCH     A,AEX
        RET                 ;RETURN
```

DOUBLE RIGHT LOGICAL SHIFT

```
RLSH:   XCH     A,AEX       ;SHIFT AEX
        RRC     A
        XCH     A,AEX
        RRC     A           ;SHIFT A
        RET                 ;RETURN
```

DOUBLE RIGHT ARITHMETIC SHIFT

```
RASH:   CLR     C           ;SET CARRY
        CPL     C

        XCH     A,AEX       ;IF AEX[7]<>1 THEN
        JB7     $+3
        CLR     C           ;CLEAR CARRY
        RRC     A           ;SHIFT C INTO AEX
        XCH     A,AEX
        RRC     A           ;SHIFT A
        RET                 ;RETURN
```

SINGLE PRECISION BINARY MULTIPLY

This routine assumes a one-byte multiplier and a one-byte multiplicand. The product, therefore, is two-bytes long.

The algorithm follows these steps:

1. The registers are arranged as follows:

 ACC — 0
 R1 — Multiplier
 R2 — Multiplicand
 R3 — Loop Counter (=8)

 The Accumulator and register R1 are treated as a register pair when they are shifted right (see Step 2)

2. The Accumulator and R1 are shifted right one place, thus the LSB of the multiplier goes into the carry.
3. The multiplicand is added to the accumulator if the carry bit is a 'one'. No action if the carry is a 'zero'.
4. Decrement the loop counter and loop (return to Step 2) until it reaches zero.
5. Shift the result right one last time just before exiting the routine

*The result will be found in the Accumulator (MS Byte) and R1 (LS Byte).

BINARY MULTIPLY

```
BMPY:   MOV   R3,#08H  ;SET COUNTER TO 8
        CLR   A        ;CLEAR A
        CLR   C        ;CLEAR CARRY BIT
BMPI:   RRC   A        ;DOUBLE SHIFT RIGHT ACC & R1
        XCH   A,R1     ;INTO CARRY
        RRC   A
        XCH   A,R1
        JNC   BMP3     ;IF CARRY=1 ADD, OTHERWISE DON'T
        ADD   A,R2     ;ADD MULTIPLICAND TO ACCUMULATOR
BMP3:   DJNZ  R3,BMPI  ;DECREMENT COUNTER AND LOOP IF 0
        RRC   A        ;DO A FINAL RIGHT SHIFT AT THE
        XCH   A,R1     ;END OF THE ROUTINE
        RRC   A
        XCH   A,R1
```

INTERRUPT HANDLING

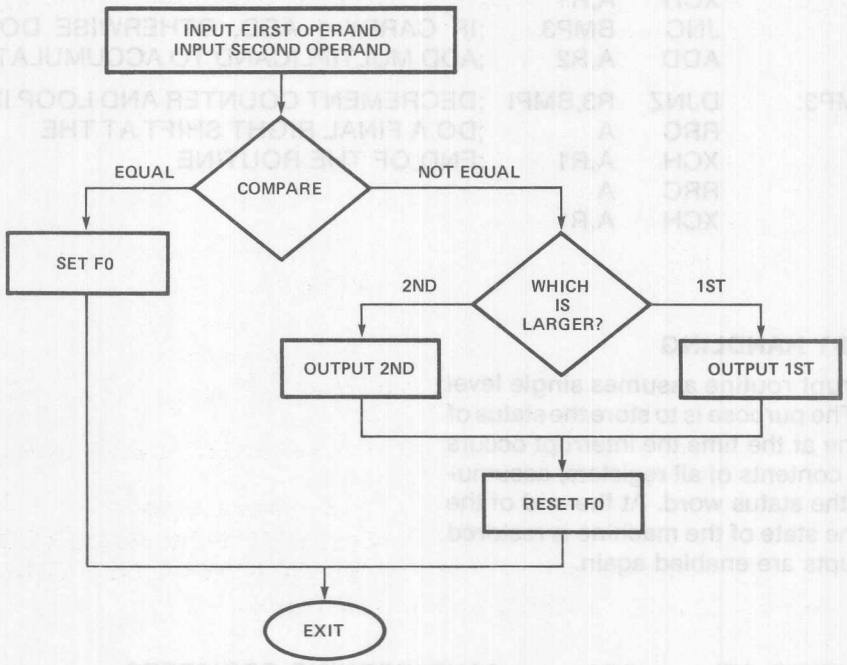
This interrupt routine assumes single level interrupt. The purpose is to store the status of the machine at the time the interrupt occurs by storing contents of all registers, accumulator, and the status word. At the end of the interrupt the state of the machine is restored and interrupts are enabled again.

```
INTRPT: SEL   RB1      ;SAVE WORKING REGISTERS
        MOV   @R0,A    ;R0 IN ALTERNATE REGISTER
                       ;BANK CONTAINS SACC
                       ;POINTER FOR SAVING
                       ;ACCUMULATOR
          |     |      {INTERRUPT SERVICE
          |     |      {ROUTINE
        MOV   R0,SACC  ;RESTORE SACC
        MOV   A,@R0    ;RESTORE ACCUMULATOR
        RETR           ;RESTORE WORKING REGISTERS
                       ;RESTORE PSW AND
                       ;RE-ENABLE INTERRUPTS
```

2 BYTE PROCESSING SYSTEM

A suggested model of a processing routine takes two single byte inputs from different ports, compares them, and performs the following, depending on the result of the comparison:

(If Equal) Sets Flag and Exits
(If Not Equal) Resets Flag and Outputs the Larger to a Third Port




```
PROCESS: CLR   F0      ;CLEAR F0 BIT (INITIALIZE)
         IN    A,P1    ;READ FIRST INPUT, STORE IN R0
         MOV   R0,A
         IN    A,P2    ;READ SECOND INPUT, STORE IN R1
         MOV   R1,A
         CPL   A       ;SUBTRACT SECOND FROM FIRST
         INC   A       ;(2's COMPLEMENT AND ADD)
         ADD   A,R0
         JNC   EQUL    ;BRANCH IF THEY ARE EQUAL
         JB7   SECOND  ;IF NEGATIVE, SECOND WAS LARGER
         MOV   A,R0    ;ELSE, OUTPUT FIRST
         OUTL  BUS,A
         JMP   DONE    ;EXIT

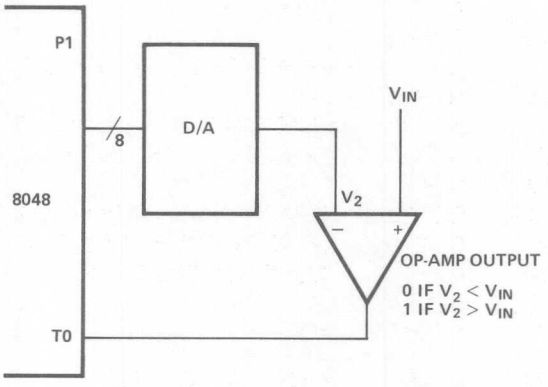
SECOND:  MOV   A,R1    ;OUTPUT SECOND
         OUTL  BUS,A
         JMP   DONE    ;EXIT
EQUL:    CPL   F0      ;SET F0
         JMP   DONE    ;EXIT
```

A/D CONVERTER

An A/D converter can be constructed from a D/A converter, a comparator op-amp and a short software routine that performs successive approximation.

The processor sends 8-bits of data out to the DAC via an output port. The output of the DAC is compared to the analog input being converted. The result of the comparison (0 if lower, 1 if higher) then goes back into the processor for handling either via an input port or an input line that sets a flag. This all allows the processor to estimate the proper digital representation of the analog input by first typing the MSB — and keeping it if the input says 'too low still' or dropping it if the input says 'too high now'. From there each bit in order of significance is tried and either kept or discarded.

```
        MOV     R7,#08H  ;COUNTER R7=8
        CLR     A        ;CLEAR A, R5, R6
        MOV     R5,A
        MOV     R6,A
        CLR     C        ;SET CARRY
        CPL     C
LOOP:   MOV     A,R5     ;MOVE TEST BIT RIGHT
        RRC     A        :FROM MSB TO LSB
        MOV     R5,A
        ORL     A,R6     ;ADD IT TO PRESENT VALUE IN R6
        OUTL    P1,A
        JT0     NOPE     ;TEST THAT NEW VALUE
                         ;IF FLAG IS HIGH NEW VALUE TOO LARGE
        MOV     R6,A     ;IF FLAG LOW, NEW VALUE RETAINED
NOPE:   DJNZ    R7,LOOP  ;GO ON TO NEXT BIT
```



Chapter 6

MCS-48™ COMPONENT SPECIFICATIONS



MCS-48™ COMPONENT SPECIFICATIONS

8048	ROM Microcomputer	6-1
8648	EPROM Microcomputer	6-1
8748	EPROM Microcomputers	6-1
8035	Microcomputers	6-1
8049/8039/ 8039-6	Microcomputers	6-9
8021	Microcomputer	6-15
8022	Microcomputer	6-21
8355	ROM and I/O Expander	6-33
8755	EPROM and I/O Expander	6-39
8155/8156	RAM and I/O Expander	6-45
8243	MCS-48™ I/O Expander	6-57

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

8048/8648/8748/8035
SINGLE COMPONENT 8-BIT MICROCOMPUTER

- 8048 Mask Programmable ROM
- 8648 One-Time Factory Programmable EPROM
- 8748 User Programmable/Erasable EPROM
- 8035 External ROM or EPROM

- 8-Bit CPU, ROM, RAM, I/O in Single Package
- Interchangeable ROM and EPROM Versions
- Single 5V Supply
- 2.5 μsec and 5.0 μsec Cycle Versions: All Instructions 1 or 2 Cycles
- Over 90 Instructions: 70% Single Byte
- 1K x 8 ROM/EPROM 64 x 8 RAM 27 I/O Lines
- Interval Timer/Event Counter
- Easily Expandable Memory and I/O
- Compatible with 8000 Series Peripherals
- Single Level Interrupt

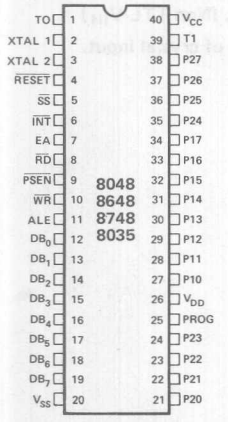
The Intel® 8048/8648/8748/8035 is a totally self-sufficient 8-bit parallel computer fabricated on a single silicon chip using Intel's N-channel silicon gate MOS process.

The 8048 contains a 1K x 8 program memory, a 64 x 8 RAM data memory, 27 I/O lines, and an 8-bit timer/counter in addition to onboard oscillator and clock circuits. For systems that require extra capability, the 8048 can be expanded using standard memories and MCS-80™ (8080A) peripherals. The 8035 is the equivalent of an 8048 without program memory. The 8035L has the RAM power down mode of the 8048 while the 8035 does not.

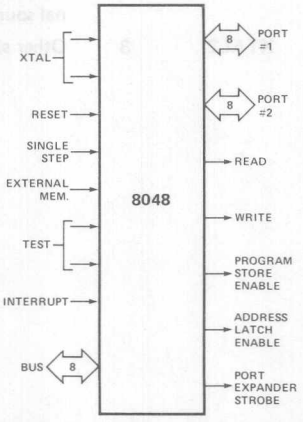
To reduce development problems to a minimum and provide maximum flexibility, three interchangeable pin-compatible versions of this single component microcomputer exist: the 8748 with user-programmable and erasable EPROM program memory for prototype and preproduction systems, the 8048 with factory-programmed mask ROM program memory for low cost, high volume production, and the 8035 without program memory for use with external program memories.

This microprocessor is designed to be an efficient controller as well as an arithmetic processor. The 8048 has extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over 2 bytes in length.

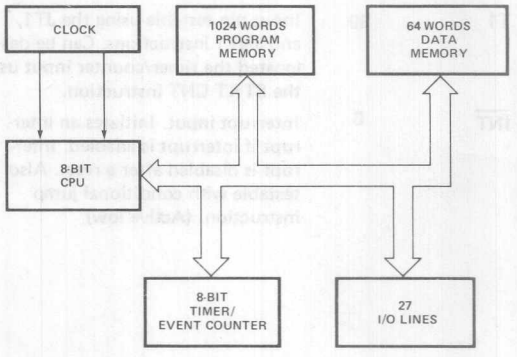
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



V_{SS}	20	Circuit GND potential
V_{DD}	26	Programming power supply; +25V during program, +5V during operation for both ROM and PROM. Low power standby pin in 8048 ROM version.
V_{CC}	40	Main power supply; +5V during operation and programming.
PROG	25	Program pulse (+25V) input pin during 8748 programming.
		Output strobe for 8243 I/O expander.
P10-P17 Port 1	27-34	8-bit quasi-bidirectional port.
P20-P27 Port 2	21-24 35-38	8-bit quasi-bidirectional port. P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243.
DB_0-DB_7 BUS	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of $\overline{PSEN}$. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction. T0 is also used during programming.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low)
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the bus from an external device.
		Used as a read strobe to external data memory. (Active low)
$\overline{RESET}$	4	Input which is used to initialize the processor. Also used during PROM programming verification, and power down. (Active low) (Non TTL V_{IH})
$\overline{WR}$	10	Output strobe during a bus write. (Active low)
		Used as write strobe to external data memory.
ALE	11	Address latch enable. This signal occurs once during each cycle and is useful as a clock output.
		The negative edge of ALE strobes address into external data and program memory.
$\overline{PSEN}$	9	Program store enable. This output occurs only during a fetch to external program memory. (Active low)
$\overline{SS}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction. (Active low)
EA	7	External access input which forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active high)
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Non TTL V_{IH})
XTAL2	3	Other side of crystal input.

INSTRUCTION SET

	Mnemonic	Description	Bytes	Cycle
Accumulator	ADD A, R	Add register to A	1	1
	ADD A, @R	Add data memory to A	1	1
	ADD A, #data	Add immediate to A	2	2
	ADDC A, R	Add register with carry	1	1
	ADDC A, @R	Add data memory with carry	1	1
	ADDC A, #data	Add immediate with carry	2	2
	ANL A, R	And register to A	1	1
	ANL A, @R	And data memory to A	1	1
	ANL A, #data	And immediate to A	2	2
	ORL A, R	Or register to A	1	1
	ORL A, @R	Or data memory to A	1	1
	ORL A, #data	Or immediate to A	2	2
	XRL A, R	Exclusive or register to A	1	1
	XRL A, @R	Exclusive or data memory to A	1	1
	XRL A, #data	Exclusive or immediate to A	2	2
	INC A	Increment A	1	1
	DEC A	Decrement A	1	1
	CLR A	Clear A	1	1
	CPL A	Complement A	1	1
	DA A	Decimal adjust A	1	1
	SWAP A	Swap nibbles of A	1	1
	RL A	Rotate A left	1	1
	RLC A	Rotate A left through carry	1	1
	RR A	Rotate A right	1	1
	RRC A	Rotate A right through carry	1	1
Input/Output	IN A, P	Input port to A	1	2
	OUTL P, A	Output A to port	1	2
	ANL P, #data	And immediate to port	2	2
	ORL P, #data	Or immediate to port	2	2
	INS A, BUS	Input BUS to A	1	2
	OUTL BUS, A	Output A to BUS	1	2
	ANL BUS, #data	And immediate to BUS	2	2
	ORL BUS, #data	Or immediate to BUS	2	2
	MOVD A, P	Input expander port to A	1	2
	MOVD P, A	Output A to expander port	1	2
	ANLD P, A	And A to expander port	1	2
	ORLD P, A	Or A to expander port	1	2
Registers	INC R	Increment register	1	1
	INC @R	Increment data memory	1	1
	DEC R	Decrement register	1	1
Branch	JMP addr	Jump unconditional	2	2
	JMPP @A	Jump indirect	1	2
	DJNZ R, addr	Decrement register and skip	2	2
	JC addr	Jump on carry = 1	2	2
	JNC addr	Jump on carry = 0	2	2
	J Z addr	Jump on A zero	2	2
	JNZ addr	Jump on A not zero	2	2
	JT0 addr	Jump on T0 = 1	2	2
	JNT0 addr	Jump on T0 = 0	2	2
	JT1 addr	Jump on T1 = 1	2	2
	JNT1 addr	Jump on T1 = 0	2	2
	JF0 addr	Jump on F0 = 1	2	2
	JF1 addr	Jump on F1 = 1	2	2
	JTF addr	Jump on timer flag	2	2
	JNI addr	Jump on $\overline{INT}$ = 0	2	2
	JBb addr	Jump on accumulator bit	2	2

	Mnemonic	Description	Bytes	Cycles
Subroutine	CALL	Jump to subroutine	2	2
	RET	Return	1	2
	RETR	Return and restore status	1	2
Flags	CLR C	Clear carry	1	1
	CPL C	Complement carry	1	1
	CLR F0	Clear flag 0	1	1
	CPL F0	Complement flag 0	1	1
	CLR F1	Clear flag 1	1	1
	CPL F1	Complement flag 1	1	1
Data Moves	MOV A, R	Move register to A	1	1
	MOV A, @R	Move data memory to A	1	1
	MOV A, #data	Move immediate to A	2	2
	MOV R, A	Move A to register	1	1
	MOV @R, A	Move A to data memory	1	1
	MOV R, #data	Move immediate to register	2	2
	MOV @R, #data	Move immediate to data memory	2	2
	MOV A, PSW	Move PSW to A	1	1
	MOV PSW, A	Move A to PSW	1	1
	XCH A, R	Exchange A and register	1	1
	XCH A, @R	Exchange A and data memory	1	1
	XCHD A, @R	Exchange nibble of A and register	1	1
	MOVX A, @R	Move external data memory to A	1	2
	MOVX @R, A	Move A to external data memory	1	2
	MOVP A, @A	Move to A from current page	1	2
	MOVP3 A, @A	Move to A from page 3	1	2
Timer/Counter	MOV A, T	Read timer/counter	1	1
	MOV T, A	Load timer/counter	1	1
	STRT T	Start timer	1	1
	STRT CNT	Start counter	1	1
	STOP TCNT	Stop timer/counter	1	1
	EN TCNTI	Enable timer/counter interrupt	1	1
	DIS TCNTI	Disable timer/counter interrupt	1	1
Control	EN I	Enable external interrupt	1	1
	DIS I	Disable external interrupt	1	1
	SEL RB0	Select register bank 0	1	1
	SEL RB1	Select register bank 1	1	1
	SEL MB0	Select memory bank 0	1	1
	SEL MB1	Select memory bank 1	1	1
	ENT0 CLK	Enable clock output on T0	1	1
	NOP	No operation	1	1

Mnemonics copyright Intel Corporation 1976

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias 0°C to 70°C
Storage Temperature -65°C to +150°C
Voltage On Any Pin With Respect to Ground -0.5V to +7V
Power Dissipation 1.5 Watt

COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. AND OPERATING CHARACTERISTICS

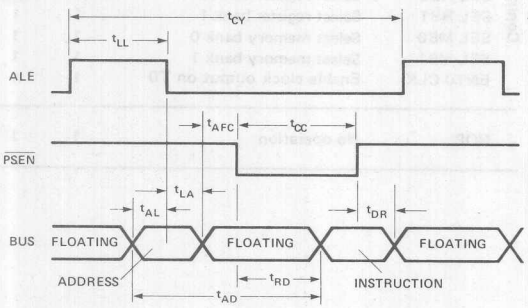
T_A = 0°C to 70°C, V_{CC} = V_{DD} = +5V ±10%*, V_{SS} = 0V

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage	-.5		.8	V	
V_{IH}	Input High Voltage (All Except XTAL1,XTAL2,$\overline{RESET}$)	2.0		V_{CC}	V	
V_{IH1}	Input High Voltage ($\overline{RESET}$, X1, X2)	3.8		V_{CC}	V	
V_{OL}	Output Low Voltage (BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)			.45	V	I_{OL} = 2.0mA
V_{OL1}	Output Low Voltage (All Other Outputs Except PROG)			.45	V	I_{OL} = 1.6mA
V_{OL2}	Output Low Voltage (PROG)			.45	V	I_{OL} = 1.0mA
V_{OH}	Output High Voltage (BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	2.4			V	I_{OH} = -100μA
V_{OH1}	Output High Voltage (All Other Outputs)	2.4			V	I_{OH} = -50μA
I_{IL}	Input Leakage Current (T1, $\overline{INT}$)			±10	μA	$V_{SS} \leq V_{IN} \leq V_{CC}$
I_{OL}	Output Leakage Current (BUS, T0) (High Impedance State)			±10	μA	$V_{SS}+.45 \leq V_{IN} \leq V_{CC}$
I_{DD}	V_{DD} Supply Current		10	20	mA	
$I_{DD} + I_{CC}$	Total Supply Current		65	135	mA	

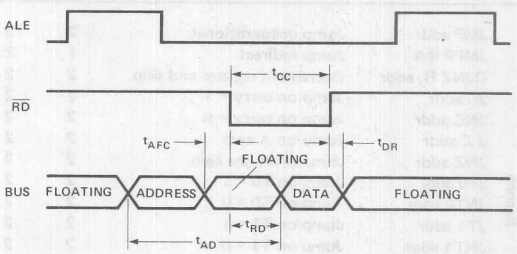
*Standard 8748 and 8035 ± 5%, ±10% available.

WAVEFORMS

Instruction Fetch From External Program Memory

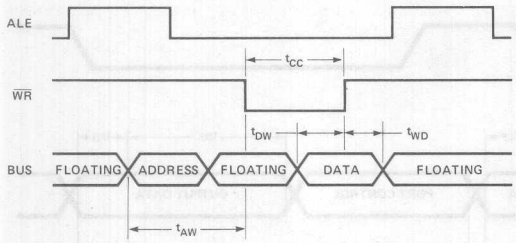


Read From External Data Memory



PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

Write to External Data Memory



A.C. CHARACTERISTICS T_A = 0°C to 70°C, V_{CC} = V_{DD} = +5V ±10%*, V_{SS} = 0V

Symbol	Parameter	8048 8648 (Note 2) 8748/8035/8035L		8748-8 8035-8		Unit	Conditions (Note 1)
		Min.	Max.	Min.	Max.		
t_{LL}	ALE Pulse Width	400		600		ns	
t_{AL}	Address Setup to ALE	150		150		ns	
t_{LA}	Address Hold from ALE	80		80		ns	
t_{CC}	Control Pulse Width ($\overline{PSEN}$, $\overline{RD}$, $\overline{WR}$)	700		1500		ns	
t_{DW}	Data Setup before $\overline{WR}$	500		640		ns	
t_{WD}	Data Hold After $\overline{WR}$	120		120		ns	C_L = 20pF
t_{CY}	Cycle Time	2.5	15.0	4.17	15.0	μs	6 MHz XTAL (3.6MHz XTAL for -8)
t_{DR}	Data Hold	0	200	0	200	ns	
t_{RD}	$\overline{PSEN}$, $\overline{RD}$ to Data In		500		750	ns	
t_{AW}	Address Setup to $\overline{WR}$	230		260		ns	
t_{AD}	Address Setup to Data In		950		1450	ns	
t_{AFC}	Address Float to $\overline{RD}$, $\overline{PSEN}$	0		0		ns	

Note 1: Control outputs: C_L = 80 pF
BUS Outputs: C_L = 150 pF, t_{CY} = 2.5μs

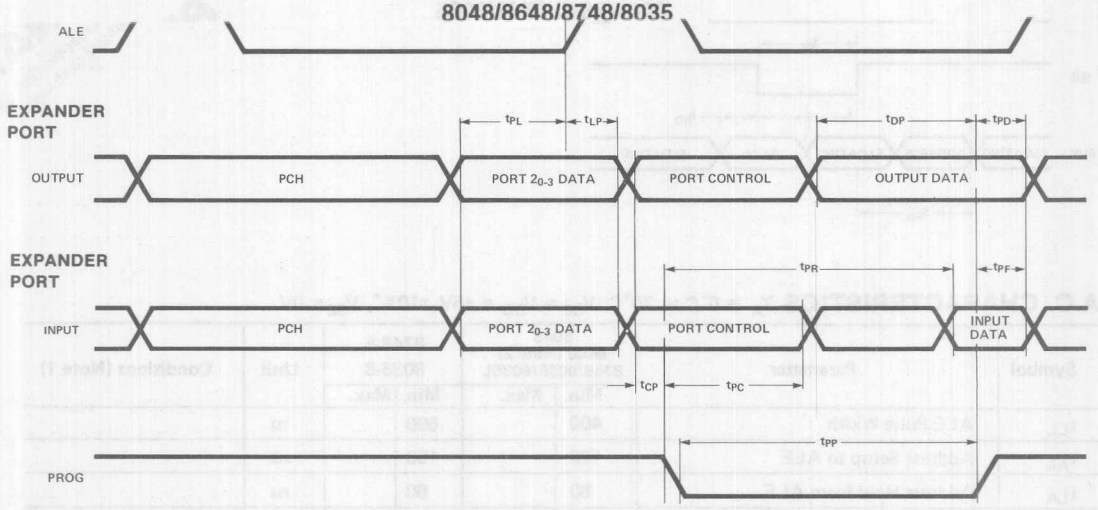
*Standard 8748 and 8035 ±5%, ±10% available.

Note 2: The 8648 is a one-time programmable (at the factory) 8748 which can be ordered as the first 25 pieces of a new 8048 ROM order. The substitution of 8648's for 8048's allows for very fast turnaround for initial code verification and evaluation units. The 8648, like the 8748, is electrically and functionally interchangeable with the 8048 with the exception of the powerdown mode which the 8648 does not support and ±5% supply tolerance instead of ±10%.

A.C. CHARACTERISTICS (PORT 2 TIMING)

T_A = 0°C to 70°C, V_{CC} = 5V±10%

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CP}	Port Control Setup Before Falling Edge of $\overline{PROG}$	110		ns	
t_{PC}	Port Control Hold After Falling Edge of $\overline{PROG}$	140		ns	
t_{PR}	PROG to Time P2 Input Must Be Valid		810	ns	
t_{DP}	Output Data Setup Time	220		ns	
t_{PD}	Output Data Hold Time	65		ns	
t_{PF}	Input Data Hold Time	0	150	ns	
t_{PP}	$\overline{PROG}$ Pulse Width	1510		ns	
t_{PL}	Port 2 I/O Data Setup	400		ns	
t_{LP}	Port 2 I/O Data Hold	150		ns	



PROGRAMMING, VERIFYING, AND ERASING THE 8748 EPROM

Programming Verification

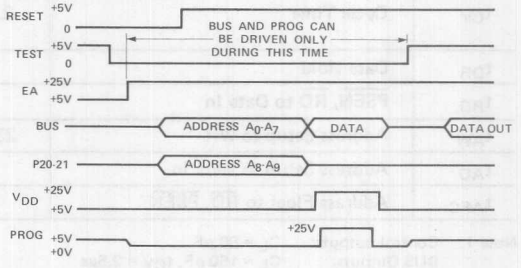
In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock Input (1 to 6MHz)
Reset	Initialization and Address Latching
Test 0	Selection of Program of Verify Mode
EA	Activation of Program/Verify Modes
BUS	Address and Data Input Data Output During Verify
P20-1	Address Input
V_{DD}	Programming Power Supply
PROG	Program Pulse Input

WARNING:

An attempt to program a missocketed 8748 will result in severe damage to the part. An indication of a properly socketed part is the appearance of the ALE clock output. The lack of this clock may be used to disable the programmer.

Programming/Verification Sequence



The Program/Verify sequence is:

1. V_{DD} = 5v, Clock applied or internal oscillator operating, RESET = 0v, TEST 0 = 5v, EA = 5v, BUS and PROG floating.
2. Insert 8748 in programming socket
3. TEST 0 = 0v (select program mode)
4. EA = 25v (activate program mode)
5. Address applied to BUS and P20-1
6. RESET = 5v (latch address)
7. Data applied to BUS
8. V_{DD} = 25v (programming power)
9. PROG = 0v followed by one 50ms pulse to 25v
10. V_{DD} = 5v
11. TEST 0 = 5v (verify mode)
12. Read and verify data on BUS
13. TEST 0 = 0v
14. RESET = 0v and repeat from step 5
15. Programmer should be at conditions of step 1 when 8748 is removed from socket.

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

AC TIMING SPECIFICATION FOR PROGRAMMING

$T_A = 25°C \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{DD} = 25V \pm 1V$

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	Address Setup Time to $\overline{RESET}$ ↑	4tcy			
t_{WA}	Address Hold Time After $\overline{RESET}$ ↑	4tcy			
t_{DW}	Data in Setup Time to PROG ↑	4tcy			
t_{WD}	Data in Hold Time After PROG ↓	4tcy			
t_{PH}	$\overline{RESET}$ Hold Time to Verify	4tcy			
t_{VDDW}	V_{DD}	4tcy			
t_{VDDH}	V_{DD} Hold Time After PROG ↓	0			
t_{PW}	Program Pulse Width	50	60	MS	
t_{TW}	Test 0 Setup Time for Program Mode	4tcy			
t_{WT}	Test 0 Hold Time After Program Mode	4tcy			
t_{DO}	Test 0 to Data Out Delay		4tcy		
t_{WW}	$\overline{RESET}$ Pulse Width to Latch Address	4tcy			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	2.0	μs	
t_{CY}	CPU Operation Cycle Time	5.0		μs	
t_{RE}	$\overline{RESET}$ Setup Time Before EA ↑,	4tcy			

Note: If Test 0 is high t_{DO} can be triggered by $\overline{RESET}$ ↑.

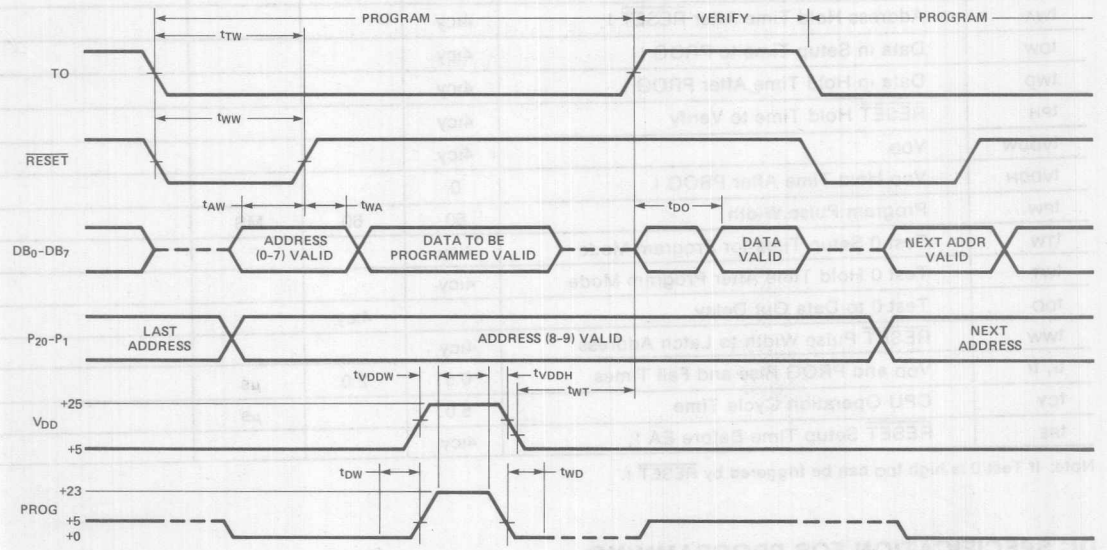
DC SPECIFICATION FOR PROGRAMMING

$T_A = 25°C \pm 5°C$, $V_{CC} = 5V \pm 5\%$, $V_{DD} = 25V \pm 1V$

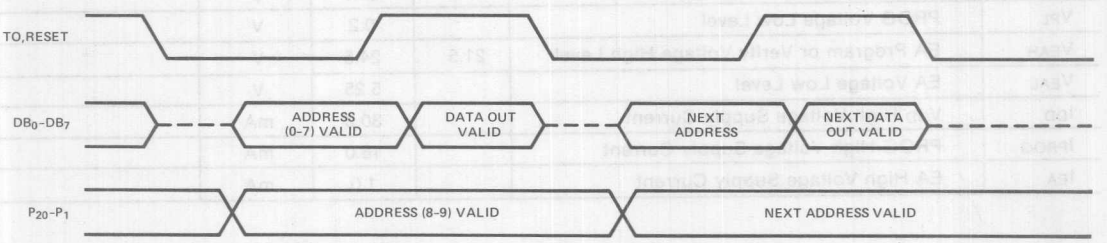
Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{DOH}	V_{DD} Program Voltage High Level	24.0	26.0	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	21.5	24.5	V	
V_{PL}	PROG Voltage Low Level		0.2	V	
V_{EAH}	EA Program or Verify Voltage High Level	21.5	24.5	V	
V_{EAL}	EA Voltage Low Level		5.25	V	
I_{DD}	V_{DD} High Voltage Supply Current		30.0	mA	
I_{PROG}	PROG High Voltage Supply Current		16.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

WAVEFORMS FOR PROGRAMMING

COMBINATION PROGRAM/VERIFY MODE (EPROM'S ONLY)



VERIFY MODE (ROM/EPROM)



NOTES:
1. PROG MUST FLOAT IF EA IS LOW (i.e., ≠ 25V), OR IF T0 = 5V FOR THE 8748. FOR THE 8048 PROG MUST ALWAYS FLOAT.
2. V_{EAH} FOR 8048 = 11.4V MIN., 12.6V MAX.
3. THE FOLLOWING CONDITIONS MUST BE MET:
 $\overline{CS}$ = TTL '1'
 A0 = TTL '0'
 THIS CAN BE DONE USING 10K RESISTORS TO V_{CC}, V_{SS} RESPECTIVELY.
4. X_1 AND X_2 DRIVEN BY 3 MHz CLOCK WILL GIVE 5μsec t_{CY}. THIS IS GOOD FOR –8 PARTS AS WELL AS NON –8 PARTS.

The 8748 EPROM can be programmed by either of two Intel products:

1. PROMPT-48 Microcomputer Design Aid, or
2. Universal PROM Programmer (UPP-101 or UPP-102) peripheral of the Intellec® Development System with a UPP-848 Personality Card.

Note: See Appendix 2 for 8048 ROM ordering procedures. To minimize turnaround time on the first 25 pieces 8648 may be specified on the ROM order.

intel®

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

NEW HIGH PERFORMANCE 8049/8039/8039-6 SINGLE COMPONENT 8-BIT MICROCOMPUTER

***8049 Mask Programmable ROM**
***8039 External ROM or EPROM**
***New 11 MHz Operation**

- **8-Bit CPU, ROM, RAM, I/O in Single Package**
- **Single 5V ± 10% Supply**
- **1.36 μsec Cycle; All Instructions 1 or 2 Cycles**
- **Over 90 Instructions: 70% Single Byte**
- **Pin Compatible with 8048/8748**
- **2K x 8 ROM**
 128 x 8 RAM
 27 I/O Lines
- **Interval Timer/Event Counter**
- **Easily Expandable Memory and I/O**
- **Compatible with MCS Memory and I/O**
- **Single Level Interrupt**

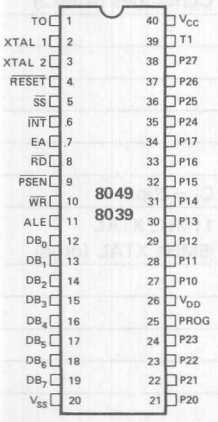
The Intel® 8049/8039/8039-6 is a totally self-sufficient 8-bit parallel computer fabricated on a single silicon chip using Intel's N-channel silicon gate MOS process.

The 8049 contains a 2K x 8 program memory, a 128 x 8 RAM data memory, 27 I/O lines, and an 8-bit timer/counter in addition to on board oscillator and clock circuits. For systems that require extra capability, the 8049 can be expanded using standard memories and MCS-80™/MCS-85™ peripherals. The 8039 is the equivalent to an 8049 without program memory. The 8039-6 is a lower speed (6MHz) version of the 8039.

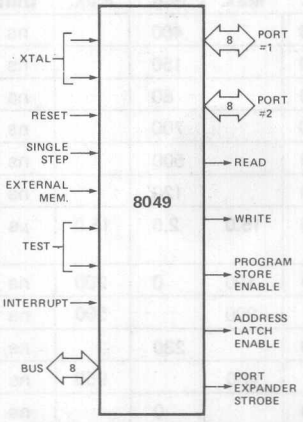
To reduce development problems to a minimum and provide maximum flexibility, two interchangeable pin-compatible versions of this single component microcomputer exist: the 8049 with factory-programmed mask ROM program memory for low-cost high volume production, and the 8039 without program memory for use with external program memories in prototype and preproduction systems.

This microprocessor is designed to be an efficient controller as well as an arithmetic processor. The 8049 has extensive bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over two bytes in length.

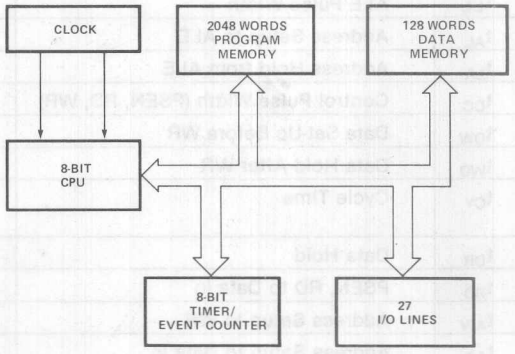
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias 0°C to 70°C
Storage Temperature -65°C to +150°C
Voltage on Any Pin With Respect to Ground -0.5V to +7V
Power Dissipation 1.5 Watt

COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{CC} = V_{DD} = +5V \pm 10\%$, $V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage	-0.5		0.8	V	
V_{IH}	Input High Voltage (All Except XTAL1, XTAL2, $\overline{RESET}$)	2.0		V_{CC}	V	
V_{IH1}	Input High Voltage ($\overline{RESET}$, X1, X2)	3.8		V_{CC}	V	
V_{OL}	Output Low Voltage (BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)			0.45	V	$I_{OL} = 2.0mA$
V_{OL1}	Output Low Voltage (All Other Outputs Except PROG)			0.45	V	$I_{OL} = 1.6mA$
V_{OL2}	Output Low Voltage (PROG)			0.45	V	$I_{OL} = 1.0mA$
V_{OH}	Output High Voltage (BUS, $\overline{RD}$, $\overline{WR}$, $\overline{PSEN}$, ALE)	2.4			V	$I_{OH} = -100\mu A$
V_{OH1}	Output High Voltage (All Other Outputs)	2.4			V	$I_{OH} = -50\mu A$
I_{IL}	Input Leakage Current (T1, $\overline{INT}$)			±10	μA	$V_{SS} \leqslant V_{IN} \leqslant V_{CC}$
I_{OL}	Output Leakage Current (Bus, T0) (High Impedance State)			±10	μA	$V_{SS} + 0.45 \leqslant V_{IN} \leqslant V_{CC}$
I_{DD}	Power Down Supply Current		25	50	mA	$T_A = 25°C$
$I_{DD}+I_{CC}$	Total Supply Current		100	170	mA	$T_A = 25°C$

A.C. CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{CC} = V_{DD} = +5V \pm 10\%$, $V_{SS} = 0V$

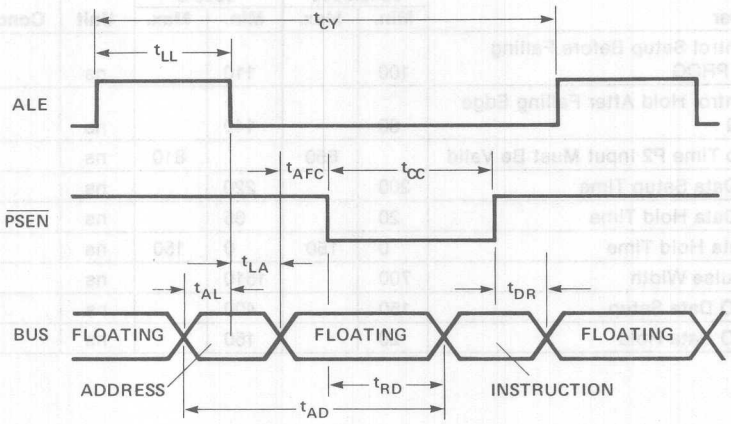
Symbol	Parameter	8049/8039 (Note 1)		8039-6		Unit	Conditions (Note 2)
		Min.	Max.	Min.	Max.		
t_{LL}	ALE Pulse Width	150		400		ns	
t_{AL}	Address Setup to ALE	70		150		ns	
t_{LA}	Address Hold from ALE	50		80		ns	
t_{CC}	Control Pulse Width (PSEN, RD, WR)	300		700		ns	
t_{DW}	Data Set-Up Before WR	250		500		ns	
t_{WD}	Data Hold After WR	40		120		ns	$C_L = 20pF$
t_{CY}	Cycle Time	1.36	15.0	2.5	15.0	μs	11MHz XTAL (6MHz XTAL for -6)
t_{DR}	Data Hold	0	100	0	200	ns	
t_{RD}	PSEN, RD to Data In		200		500	ns	
t_{AW}	Address Setup to WR	200		230		ns	
t_{AD}	Address Setup to Data In		400		950	ns	
t_{AFC}	Address Float to RD, PSEN	-10		0		ns	

Notes: 1. 8039-6 specifications are also valid for 8049/8039 operating at 6MHz.

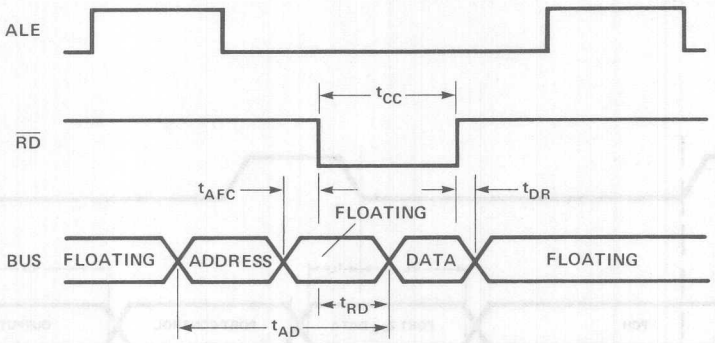
2. Control Outputs: $C_L = 80pF$
 BUS Outputs: $C_L = 150pF$

WAVEFORMS

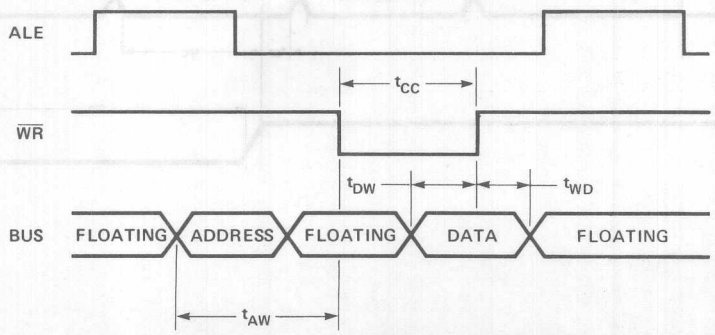
INSTRUCTION FETCH FROM EXTERNAL PROGRAM MEMORY



READ FROM EXTERNAL DATA MEMORY



WRITE TO EXTERNAL DATA MEMORY

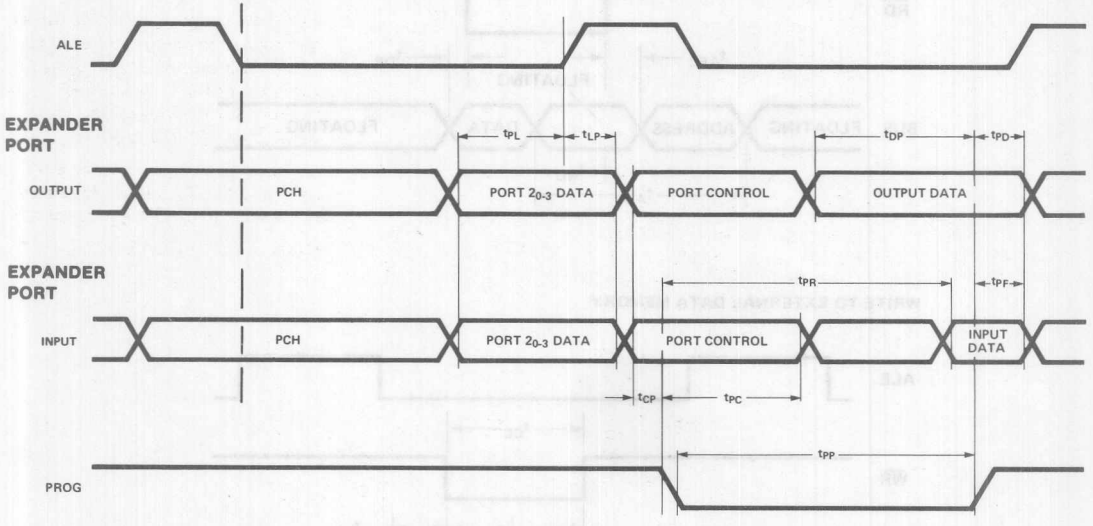


$T_A = 0°C$ to $70°C$, $V_{CC} = 5V \pm 10\%$

Symbol	Parameter	8049/8039 Min.	8049/8039 Max.	8039-6 Min.	8039-6 Max.	Unit	Conditions (Note 2)
t_{CP}	Port Control Setup Before Falling Edge of PROG	100		110		ns	
t_{PC}	Port Control Hold After Falling Edge of PROG	60		140		ns	
t_{PR}	PROG to Time P2 Input Must Be Valid		650		810	ns	
t_{DP}	Output Data Setup Time	200		220		ns	
t_{PD}	Output Data Hold Time	20		65		ns	
t_{PF}	Input Data Hold Time	0	150	0	150	ns	
t_{PP}	PROG Pulse Width	700		1510		ns	
t_{PL}	Port 2 I/O Data Setup	150		400		ns	
t_{LP}	Port 2 I/O Data Hold	20		150		ns	

WAVEFORMS

PORT 2 TIMING



PIN DESCRIPTION

Designation	Pin #	Function
V_{SS}	20	Circuit GND potential
V_{DD}	26	+5V during operation. Low power standby pin.
V_{CC}	40	Main power supply; +5V during operation.
PROG	25	Output strobe for 8243 I/O expander.
P10-P17 Port 1	27-34	8-bit quasi-bidirectional port.
P20-P27 Port 2	21-24 35-38	8-bit quasi-bidirectional port. P20-P23 contain the four high order program counter bits during an external program memory fetch and serve as a 4-bit I/O expander bus for 8243
D0-D7 BUS	12-19	True bidirectional port which can be written or read synchronously using the $\overline{RD}$, $\overline{WR}$ strobes. The port can also be statically latched. Contains the 8 low order program counter bits during an external program memory fetch, and receives the addressed instruction under the control of PSEN. Also contains the address and data during an external RAM data store instruction, under control of ALE, $\overline{RD}$, and $\overline{WR}$.
T0	1	Input pin testable using the conditional transfer instructions JT0 and JNT0. T0 can be designated as a clock output using ENT0 CLK instruction.
T1	39	Input pin testable using the JT1, and JNT1 instructions. Can be designated the timer/counter input using the STRT CNT instruction.
$\overline{INT}$	6	Interrupt input. Initiates an interrupt if interrupt is enabled. Interrupt is disabled after a reset. Also testable with conditional jump instruction. (Active low)
$\overline{RD}$	8	Output strobe activated during a BUS read. Can be used to enable data onto the BUS from an external device. Used as a Read Strobe to External Data Memory. (Active low)
$\overline{RESET}$	4	Input which is used to initialize the processor. Also used during verification, and power down. (Active low) (Non TTL V_{IH})
$\overline{WR}$	10	Output strobe during a BUS write. (Active low) Used as write strobe to External Data Memory.
ALE	11	Address Latch Enable. This signal occurs once during each cycle and is useful as a clock output. The negative edge of ALE strobes address into external data and program memory.
$\overline{PSEN}$	9	Program Store Enable. This output occurs only during a fetch to external program memory. (Active low)
$\overline{SS}$	5	Single step input can be used in conjunction with ALE to "single step" the processor through each instruction. (Active low)
EA	7	External Access input which forces all program memory fetches to reference external memory. Useful for emulation and debug, and essential for testing and program verification. (Active high)
XTAL1	2	One side of crystal input for internal oscillator. Also input for external source. (Not TTL Compatible)
XTAL2	3	Other side of crystal input.

INSTRUCTION SET

	Mnemonic	Description	Bytes	Cycle
Accumulator	ADD A, R	Add register to A	1	1
	ADD A, @R	Add data memory to A	1	1
	ADD A, #data	Add immediate to A	2	2
	ADDC A, R	Add register with carry	1	1
	ADDC A, @R	Add data memory with carry	1	1
	ADDC A, #data	Add immediate with carry	2	2
	ANL A, R	And register to A	1	1
	ANL A, @R	And data memory to A	1	1
	ANL A, #data	And immediate to A	2	2
	ORL A, R	Or register to A	1	1
	ORL A, @R	Or data memory to A	1	1
	ORL A, #data	Or immediate to A	2	2
	XRL A, R	Exclusive Or register to A	1	1
	XRL A, @R	Exclusive or data memory to A	1	1
	XRL A, #data	Exclusive or immediate to A	2	2
	INC A	Increment A	1	1
	DEC A	Decrement A	1	1
	CLR A	Clear A	1	1
	CPL A	Complement A	1	1
	DA A	Decimal Adjust A	1	1
	SWAP A	Swap nibbles of A	1	1
	RL A	Rotate A left	1	1
	RLC A	Rotate A left through carry	1	1
	RR A	Rotate A right	1	1
	RRC A	Rotate A right through carry	1	1
Input/Output	IN A, P	Input port to A	1	2
	OUTL P, A	Output A to port	1	2
	ANL P, #data	And immediate to port	2	2
	ORL P, #data	Or immediate to port	2	2
	INS A, BUS	Input BUS to A	1	2
	OUTL BUS, A	Output A to BUS	1	2
	ANL BUS, #data	And immediate to BUS	2	2
	ORL BUS, #data	Or immediate to BUS	2	2
	MOVD A, P	Input Expander port to A	1	2
	MOVD P, A	Output A to Expander port	1	2
	ANLD P, A	And A to Expander port	1	2
	ORLD P, A	Or A to Expander port	1	2
Registers	INC R	Increment register	1	1
	INC @R	Increment data memory	1	1
	DEC R	Decrement register	1	1
Branch	JMP addr	Jump unconditional	2	2
	JMPP @A	Jump indirect	1	2
	DJNZ R, addr	Decrement register and skip	2	2
	JC addr	Jump on Carry = 1	2	2
	JNC addr	Jump on Carry = 0	2	2
	J Z addr	Jump on A Zero	2	2
	JNZ addr	Jump on A not Zero	2	2
	JT0 addr	Jump on T0 = 1	2	2
	JNT0 addr	Jump on T0 = 0	2	2
	JT1 addr	Jump on T1 = 1	2	2
	JNT1 addr	Jump on T1 = 0	2	2
	JF0 addr	Jump on F0 = 1	2	2
	JF1 addr	Jump on F1 = 1	2	2
	JTF addr	Jump on timer flag	2	2
	JNI addr	Jump on $\overline{INT}$ = 0	2	2
	JBb addr	Jump on Accumulator Bit	2	2

	Mnemonic	Description	Bytes	Cycles
Subroutine	CALL	Jump to subroutine	2	2
	RET	Return	1	2
	RETR	Return and restore status	1	2
Flags	CLR C	Clear Carry	1	1
	CPL C	Complement Carry	1	1
	CLR F0	Clear Flag 0	1	1
	CPL F0	Complement Flag 0	1	1
	CLR F1	Clear Flag 1	1	1
	CPL F1	Complement Flag 1	1	1
Data Moves	MOV A, R	Move register to A	1	1
	MOV A, @R	Move data memory to A	1	1
	MOV A, #data	Move immediate to A	2	2
	MOV R, A	Move A to register	1	1
	MOV @R, A	Move A to data memory	1	1
	MOV R, #data	Move immediate to register	2	2
	MOV @R, #data	Move immediate to data memory	2	2
	Move A, PSW	Move PSW to A	1	1
	MOV PSW, A	Move A to PSW	1	1
	XCH A, R	Exchange A and register	1	1
	XCHA,@R	Exchange A and data memory	1	1
	XCHD A, @R	Exchange nibble of A and register	1	1
	MOVX A, @R	Move external data memory to A	1	2
	MOVX @R, A	Move A to external data memory	1	2
	MOVP A, @A	Move to A from current page	1	2
	MOVP3 A, @A	Move to A from Page 3	1	2
Timer/Counter	MOV A, T	Read Timer/Counter	1	1
	MOV T, A	Load Timer/Counter	1	1
	STRT T	Start Timer	1	1
	STRT CNT	Start Counter	1	1
	STOP TCNT	Stop Timer/Counter	1	1
	EN TCNTI	Enable Timer/Counter Interrupt	1	1
	DIS TCNTI	Disable Timer/Counter Interrupt	1	1
Control	EN I	Enable external interrupt	1	1
	DIS I	Disable external interrupt	1	1
	SEL RB0	Select register bank 0	1	1
	SEL RB1	Select register bank 1	1	1
	SEL MB0	Select memory bank 0	1	1
	SEL MB1	Select memory bank 1	1	1
	ENT0 CLK	Enable Clock output on T0	1	1
	NOP	No Operation	1	1

Mnemonics copyright Intel Corporation 1976, 1977, 1978

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

8021
SINGLE COMPONENT 8-BIT MICROCOMPUTER

- **8-Bit CPU, ROM, RAM, I/O in Single 28-Pin Package**
- **Single 5V Supply (+4.5V to 6.5V)**
- **10 μsec Cycle; All Instructions 1 or 2 Cycles**
- **Instructions —8748 Subset**
- **High Current Drive Capability—2 Pins**
- **1K × 8 ROM**
 64 × 8 RAM
 21 I/O Lines
- **Interval Timer/Event Counter**
- **Clock Generated With Single Resistor or Inductor**
- **Zero-Cross Detection Capability**
- **Easily Expandable I/O**

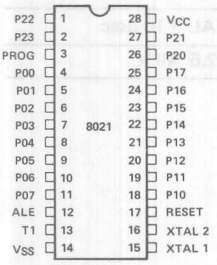
The Intel® 8021 is a totally self-sufficient 8-bit parallel computer fabricated on a single silicon chip using Intel's N-channel silicon gate MOS process. The features of the 8021 include a subset of the 8048 optimized for low cost, high volume applications, plus additional I/O flexibility and power.

The 8021 contains a 1K X 8 program memory, a 64 X 8 data memory, 21 I/O lines, and an 8-bit timer/event counter, in addition to on-board oscillator and clock circuits. For systems that require extra I/O capability, the 8021 can be expanded using the 8243 or discrete logic.

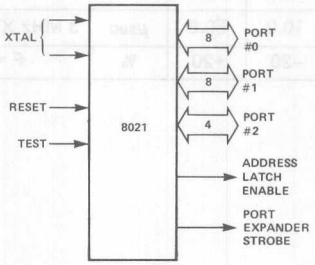
This microprocessor is designed to be an efficient controller as well as an arithmetic processor. The 8021 has bit handling capability as well as facilities for both binary and BCD arithmetic. Efficient use of program memory results from an instruction set consisting mostly of single byte instructions and no instructions over two bytes in length.

To minimize development problems and maximize flexibility, an 8021 system can be easily designed using the 8021 emulation board, EMB-21. The EMB-21 contains a 40-pin socket which can accommodate either the 8748 shipped with the board or an ICE-48 plug. Also, the necessary discrete logic to reproduce the 8021's additional I/O features is included.

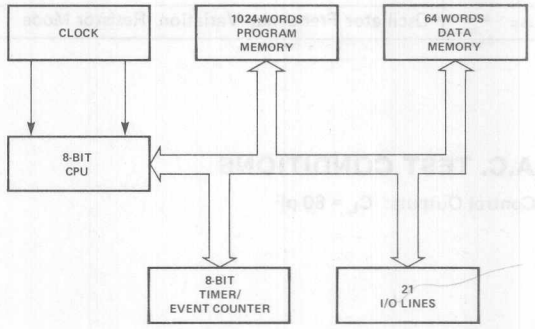
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias. 0°C to 70°C
Storage Temperature -65°C to +150°C
Voltage on Any Pin with
Respect to Ground -0.5V to +7V
Power Dissipation 1 W

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{CC} = 5.5V \pm 1V$, $V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage (All except XTAL1, XTAL2)	-0.5		0.8	V	
V_{IH}	Input High Voltage (All except XTAL1, XTAL2)	2.0		V_{CC}	V	$V_{CC} = 5.0V \pm 10\%$
V_{IH_1}	Input High Voltage (All except XTAL1, XTAL2)	3.0		V_{CC}	V	$V_{CC} = 5.5V \pm 1V$
V_{OL}	Output Low Voltage			0.45	V	$I_{OL} = 1.6$ mA
V_{OL_1}	Output Low Voltage (P10, P11)			2.5	V	$I_{OL} = 7$ mA
V_{OH}	Output High Voltage (All unless Open Drain)	2.4			V	$I_{OH} = 50\ \mu A$
I_{OL}	Output Leakage Current (Open Drain Option — Port 0)			±10	μA	$V_{SS} + 0.45 \leqslant V_{IN} \leqslant V_{CC}$
I_{CC}	V_{CC} Supply Current			60	mA	

A.C. CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{CC} = 5.5V \pm 1V$, $V_{SS} = 0V$

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CY}	Cycle Time	10.0	50.0	μsec	3 MHz XTAL = 10 μsec
Δ_F	Oscillator Frequency Variation -Resistor Mode	-20	+20	%	F = 2.5 MHz

A.C. TEST CONDITIONS

Control Outputs: $C_L = 80$ pF

PIN DESCRIPTION

Designation	Pin #	Function
V_{SS}	14	Circuit GND potential
V_{CC}	28	+5V power supply
PROG	3	Output strobe for 8243 I/O Expander
P00–P07 Port 0	4–11	8-bit quasi-bidirectional port
P10–P17 Port 1	18–25	8-bit quasi-bidirectional port
P20–P23 Port 2	26–27 1–2	4-bit quasi-bidirectional port P20–P23 also serve as a 4-bit I/O expander bus for 8243
T1	13	Input pin testable using the JT1 and JNT1 instructions. Can be designated the timer/event counter input using the STRT CNT instruction. Also allows zero-crossover sensing of slowly moving AC inputs.
RESET	17	Input used to initialize the processor by clearing status flip-flops and setting program counters to zero.
ALE	12	Address Latch Enable. Signal occuring once every 30 input clocks, used as an output clock.
XTAL1	15	One side of crystal, inductor, or resistor input for internal oscillator. Also input for external source. (Not TTL compatible.)
XTAL2	16	Other side of timing control element.

INSTRUCTION SET

	Mnemonic		Description	Bytes	Cycle
Accumulator	ADD	A,R	Add register to A	1	1
	ADD	A,@R	Add data memory to A	1	1
	ADD	A,#data	Add immediate to A	2	2
	ADDC	A,R	Add with carry	1	1
	ADDC	A,@R	Add with carry	1	1
	ADDC	A,#data	Add with carry	2	2
	ANL	A,R	And register to A	1	1
	ANL	A,@R	And data memory to A	1	1
	ANL	A,#data	And immediate to A	2	2
	ORL	A,R	Or register to A	1	1
	ORL	A,@R	Or data memory to A	1	1
	ORL	A,#data	Or immediate to A	2	2
	XRL	A,R	Exclusive Or register to A	1	1
	XRL	A,@R	Exclusive or data memory to A	1	1
	XRL	A,#data	Exclusive or immediate to A	2	2
	INC	A	Increment A	1	1
	DEC	A	Decrement A	1	1
	CLR	A	Clear A	1	1
	CPL	A	Complement A	1	1
	DA	A	Decimal Adjust A	1	1
	SWAP	A	Swap nibbles of A	1	1
	RL	A	Rotate A left	1	1
	RLC	A	Rotate A left through carry	1	1
	RR	A	Rotate A right	1	1
	RRC	A	Rotate A right through carry	1	1
Input/Output	IN	A,P	Input port to A	1	2
	OUTL	P,A	Output A to port	1	2
	MOVD	A,P	Input Expander port to A	1	2
	MOVD	P,A	Output A to Expander port	1	2
	ANLD	P,A	And A to Expander port	1	2
	ORLD	P,A	Or A to Expander port	1	2
Registers	INC	R	Increment register	1	1
	INC	@R	Increment data memory	1	1
Branch	JMP	addr	Jump unconditional	2	2
	JMPP	@A	Jump indirect	1	2
	DJNZ	R,addr	Decrement register and Jump on R not zero	2	2
Branch	JC	addr	Jump on Carry = 1	2	2
	JNC	addr	Jump on Carry = 0	2	2
	JZ	addr	Jump on A Zero	2	2
	JNZ	addr	Jump on A not Zero	2	2
	JT1	addr	Jump on T1 = 1	2	2
	JNT1	addr	Jump on T1 = 0	2	2
	JTF	addr	Jump on timer flag	2	2
Subroutine	CALL		Jump to subroutine	2	2
	RET		Return	1	2
Flags	CLR	C	Clear Carry	1	1
	CPL	C	Complement Carry	1	1
Data Moves	MOV	A,R	Move register to A	1	1
	MOV	A,@R	Move data memory to A	1	1
	MOV	A,#data	Move immediate to A	2	2
	MOV	R,A	Move A to register	1	1
	MOV	@R,A	Move A to data memory	1	1
	MOV	R,#data	Move immediate to register	2	2
	MOV	@R,#data	Move immediate to data memory	2	2
	XCH	A,R	Exchange A and register	1	1
	XCH	A,@R	Exchange A and data memory	1	1
	XCHD	A,@R	Exchange nibble of A and register	1	1
	MOVP	A,@A	Move to A from current page	1	2
Timer/Counter	MOV	A,T	Read Timer/Counter	1	1
	MOV	T,A	Load Timer/Counter	1	1
	STRT	T	Start Timer	1	1
	STRT	CNT	Start Counter	1	1
	STOP	TCNT	Stop Timer/Counter	1	1
	NOP		No Operation	1	1

FUNCTIONAL SPECIFICATIONS

The following is a functional description of the major elements of the 8021.

Program Memory

The 8021 contains 1K X 8 of mask programmable ROM. No external ROM expansion capability is provided.

Data Memory

A 64 X 8 dynamic RAM is located on chip for data storage. All locations are indirectly addressable and eight designated locations are directly addressable. Also, included in the memory is the address stack, addressed by a 3-bit stack pointer.

Memory is organized as shown in Figure 1. The least significant 8 addresses, 0–7, are directly addressable by any of the 11 direct register instructions. The locations are readily accessible for a variety of operations with the least number of instruction bytes required for their manipulation.

Registers 0 and 1 have yet another function, in that they can be used to indirectly address all locations in memory, using the indirect register instructions. These indirect RAM address registers, IRAR's, are especially useful for repetitive-type operations on adjacent memory locations. The indirect register instruction specifies which IRAR to use, and the contents of the IRAR is used to address a location in RAM. The contents of the addressed location is used during the execution of the instruction, and may be modified. A value larger than 63 should not be preset in the IRAR when selected by an indirect register instruction. IRAR's may point to addresses 0–7, if desired.

Locations 8–23 may be used as the address stack. The address stack enables the processor to keep track of the return addresses generated from CALL instructions. A 3-bit stack pointer (SP) supplies the address of the locations to be loaded with the next return address generated. The SP to this pushdown stack is incremented by one after a return address is stored, and decremented by one before an address is fetched during a RET. A total of 8 levels of nesting is possible. The SP is initialized to location 8 upon RESET. Since each address is 10 bits long, two bytes must be used to store a single address. The SP is incremented and decremented by one, but each increment or decrement moves the address pointed to by two. Therefore, only even numbered addresses are pointed to.

If a particular application does not require 8 levels of nesting, the unused portion of the stack may be used as any other indirectly addressable scratchpad location. For example, if only 3 levels of subroutine nesting are used, then only locations 8–15 need be reserved for the address stack, and locations 16–63 can be used for data storage. The actual program counter address is not stored in the address stack. A separate register retains its value.

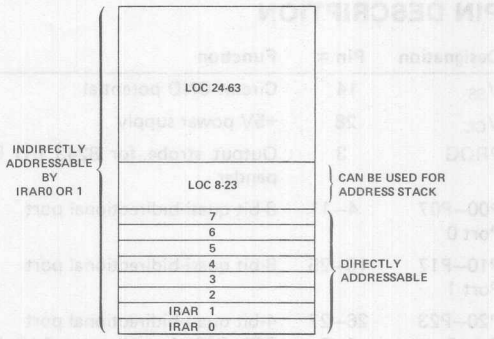


Figure 1. Internal RAM Organization

Oscillator and Clock

The 8021 contains its own onboard oscillator and clock circuit, requiring only an external timing control element. This control element can be a crystal, inductor, resistor, or clock in. The capacitor normally required in resistor or inductor timing control operation is integrated onto the 8021. All internal time slots are derived from the external element, and all outputs are a function of the oscillator frequency. Pins X1 and X2 are used to input the particular control element. An instruction cycle consists of 10 states, and each state is a time slot of 3 oscillator periods. Therefore, to obtain a 10 μsec instruction cycle, a 3 MHz crystal should be used.

Timer/Event Counter

An interval timer is available to enable the user to keep track of time elapsed or number of events occurred, during normal program execution and flow.

By a MOV T,A instruction, the contents of the accumulator are loaded to the timer. At the STRT T command an internal prescaler is zeroed and thereafter increments once each 30 input clocks (once each single cycle instruction, twice each double cycle instruction). The prescaler is a divide by 32. At the (11111) to (00000) transition the timer is incremented. The timer is 8 bits and an overflow (FFH) to (00H) timer flag is set. A conditional branch is available for testing this flag, the flag being reset each test. Total count capacity for the timer is $2^8 \times 2^5 = 8192$ or 81.9 msec at a 10 μsec cycle time. Contents of the timer are moved to the accumulator by the MOV A,T instruction without disturbing the counting process.

The timer may also be used as an event counter. After a STRT CNT command, the chip will respond to a high to low transition on the Test 1 pin by incrementing the timer. Transitions can occur no faster than one each three instruction cycles.

The timer and event functions are exclusive. Counting or timing may be started or stopped (STOP TCNT) at will.

Input/Output Capabilities

The 8021 I/O configurations are highly flexible. A number of different configurations are possible, tailoring an 8021 to a given task. Other than the power supply and dedicated pins, all other pins (20) can be used for input, output, or both, depending on the configuration.

P20–P23 and P10–P17 are quasi-bidirectional, and Test 1 is directly testable through program control. A simplified schematic of the quasi-bidirectional interface is shown in Figure 2. This configuration allows buffered outputs, and also allows external input. When writing a "0" or low value to these ports, the large pulldown device sinks an external TTL load. When writing a "1", a large current is supplied through the large pullup device to allow a fast data transfer. After a short time (less than one instruction cycle), the large device is shut off and the small pullup maintains the "1" level indefinitely. However, in this situtation, an input device capable of overriding the small amount of sustaining current supplied by the pullup device can be read. (Alternatively, the data written can be read). So, by writing a "1" to any particular pin, that pin can serve either as a true high-level latched output pin, or as just a pullup resistor on an input. This allows maximum user flexibility in selecting his input or latched output pins, with a minimum of external components.

Port 00–07 is also quasi-bidirectional, except there is no large pullup device. As outputs, this port is essentially open drain.

By mask option the small pullup devices on P00–P07 may be deleted on any pin providing a true open drain output. This is useful in driving analog circuits and certain loads, such as keyboards.

Also available is the 8243 I/O expander chip, which provides additional I/O capability with a limited number of overhead pins. This chip has 4 directly addressable 4-bit ports. It connects to the PROG pin, which provides a clock, and pins P20–P23, which provide address and data. These ports can be written with a MOVD P,A; ANLD P,A; and ORLD P,A for Ports 4–7. A high to low transition on PROG signifies that address and control are available on P20–P23. The previous data on P20–P23 before an output expander instruction is lost. Therefore, when using an output expander P20–P23 are not useful for general input/output. Reading is via the MOVD A,P. This circuit configuration is shown in Figure 3.

The Test 1 pin has a special bias input that allows zero-crossover sensing of slowly moving inputs. This is especially useful in SCR control of 60 Hz power and in developing time of day routines. As a ROM mask option there is a pullup resistor that is useful for switch contact input or standard TTL. See Figure 4.

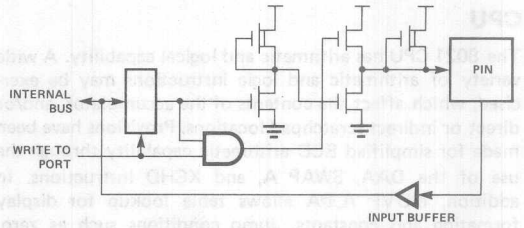


Figure 2. Quasi-Bidirectional Port Structure

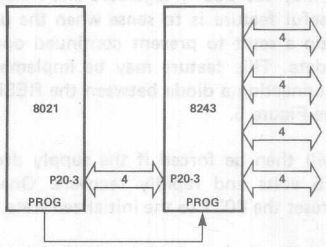


Figure 3. I/O Expander Interface

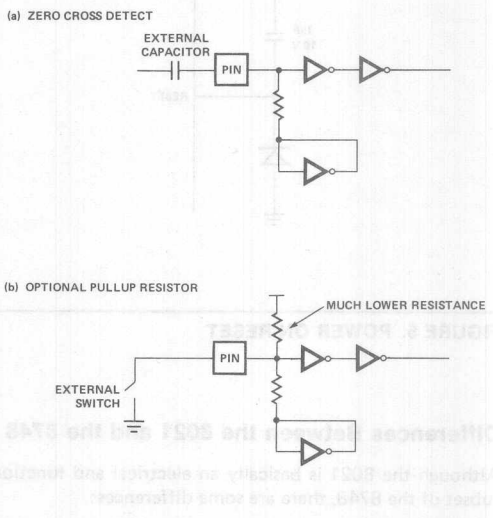


Figure 4. Test 1 Pin

CPU

The 8021 CPU has arithmetic and logical capability. A wide variety of arithmetic and logic instructions may be exercised, which affect the contents of the accumulator, and/or direct or indirect scratchpad locations. Provisions have been made for simplified BCD arithmetic capability through the use of the DAA, SWAP A, and XCHD instructions. In addition, MOVP A,@A allows table lookup for display formating and constants. Jump conditions such as zero, not zero refer to the accumulator contents at the time of the condition.

Reset

The 8021 may see poorly regulated and noisy power supplies. A useful feature is to sense when the power supply dips and do a reset to prevent continued operation with incorrect data. This feature may be implemented on the 8021 by connecting a diode between the RESET node and ground. See Figure 5.

A reset will then be forced if the supply drops approximately 1.5 volts and rapidly recovers. One instruction cycle will reset the 8021 to the initialized state.

By removing the diode and using only the capacitor, voltage drops in V_{CC} will not cause a RESET.

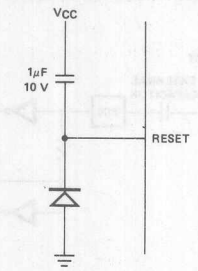


FIGURE 5. POWER ON RESET

Differences Between the 8021 and the 8748

Although the 8021 is basically an electrical and functional subset of the 8748, there are some differences:

1. *Pin Out* — As the 8021 is a 28-pin DIP, some form of adapter must be used to interface the 8021 socket to ICE-48. An emulation board, EMB-21, has been designed to perform this function. The EMB-21 also accounts for the increased flexibility of some 8021 I/O lines.
2. *Instruction Time* — The 8021 instruction cycle is 30 clock cycles long, the 8748 instruction cycle is 15 clocks long. Where exact timing is important the 8748 breadboard part should be operated at half the 8021 clock rate.
3. *Test 1* — To facilitate developing time of day routines from 60 Hz, and for SCR control, the Test 1 pin without the pullup resistor option will detect zero crossing of a capacitively coupled AC input.
4. *Quasi-Bidirectional Ports* — All 8021 ports are quasi-bidirectional to facilitate stand-alone use. Port 0 has open drain outputs and by mask option it may or may not have pullup resistors.
5. *Oscillator* — The 8021 has on-chip oscillator that is optimized for the single resistor mode. External connection will differ from the 8748.
6. *Dynamic RAM and Logic* — The 8021 utilizes dynamic RAM and some dynamic logic. Input clocking must be maintained above the minimum rate or improper operation may result.
7. *High Current Outputs* — Very high current drive is desirable for minimizing external parts required to do high power control. P10 and P11 have been designated high drive outputs capable of sinking 7 mA at V_{SS} +2.5 volts. (For clarity, this is 7 mA to V_{SS} with a 2.5 volt drop across the buffer.) These pins may, of course, be paralleled for 14 mA drive if the output logic states are always the same.
8. *Reset* — Reset has been modified on the 8021, as previously noted. A reset will be forced if the power supply drops approximately 1.5 volts and rapidly recovers, if a diode is used in the reset circuit. This prevents continued operation with incorrect data caused by a poorly regulated and/or noisy power supply.
9. *Instruction Set* — The following instructions, which are found in the 8748, have been deleted from the 8021 instruction set.

Data Moves	Registers	Branch	Timer	Control	Input/Output
MOV A,PSW	DEC R	JT0 addr	EN TCNTI	EN I	ANL P,#data
MOV PSW,A		JNT0 addr	DIS TCNTI	DIS I	ORL P,#data
MOVX A,@R	**Flags**	JF0 addr		SEL RB0	INS A,BUS *
MOVX @R,A		JF1 addr	**Subroutine**	SEL RB1	OUTL BUS,A *
MOVP3 A,@A	CLR F0	JNI addr		SEL MB0	ANL BUS,#data
	CPL F0	JBb addr	RETR	SEL MB1	ORL BUS,#data
	CLR F1			ENT0 CLK	
	CPL F1				

*These Instructions have been replaced in the 8021 by IN A,PO and OUTL PO,A respectively.

8022
SINGLE COMPONENT 8-BIT MICROCOMPUTER WITH ON-CHIP A/D CONVERTER

PRELIMINARY
Notice: This is not a final specification. Some parameter limits are subject to change.

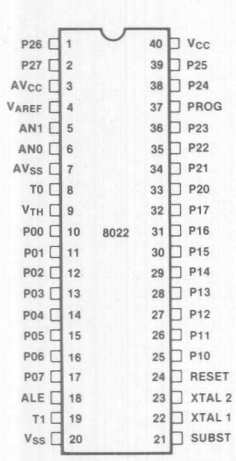
- 8-Bit CPU, ROM, RAM, I/O in Single 40-Pin Package
- On-Chip 8-Bit A/D Converter; Two Input Channels
- 8 Comparator Inputs (Port 0)
- Zero-Cross Detection Capability
- Single 5V Supply (4.5V to 6.5V)
- High Current Drive Capability — 2 Pins
- Two Interrupts — External and Timer
- 2K x 8 ROM, 64 x 8 RAM, 28 I/O Lines
- 10 μsec Cycle; All Instructions 1 or 2 Cycles
- Instructions — 8048 Subset
- Interval Timer/Event Counter
- Clock Generated with Single Resistor, Inductor, or Crystal
- Easily Expandable I/O

The Intel® 8022 is the newest member of the MCS-48™ family of single chip 8-bit microcomputers. It is designed to satisfy the requirements of low cost, high volume applications which involve analog signals, capacitive touchpanel keyboards, and/or large ROM space. The 8022 addresses these applications by integrating many new functions on-chip, such as A/D conversion, comparator inputs and zero-cross detection.

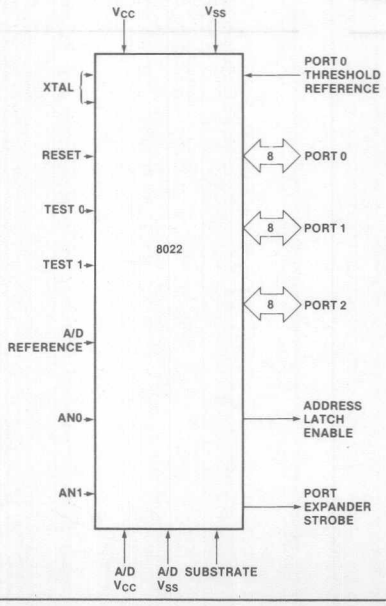
The features of the 8022 include 2K bytes of program memory (ROM), 64 bytes of data memory (RAM), 28 I/O lines, an on-chip 8-bit A/D converter with two input channels, an 8-bit port with comparator inputs for interfacing to low voltage capacitive touchpanels or other non-TTL interfaces, external and timer interrupts, and zero-cross detection capability. In addition, it contains the 8-bit interval timer/event counter, on-board oscillator and clock circuitry, single 5V power supply requirement, and easily expandable I/O structure common to all members of the MCS-48 family.

The 8022 is designed to be an efficient controller as well as an arithmetic processor. It has bit handling capability plus facilities for both binary and BCD arithmetic. Efficient use of program memory results from using the MCS-48 instruction set which consists mostly of single byte instructions and has extensive conditional jump and direct table lookup capability. Program memory usage is further reduced via the 8022's hardware implementation of the A/D converter which simplifies interfacing to analog signals.

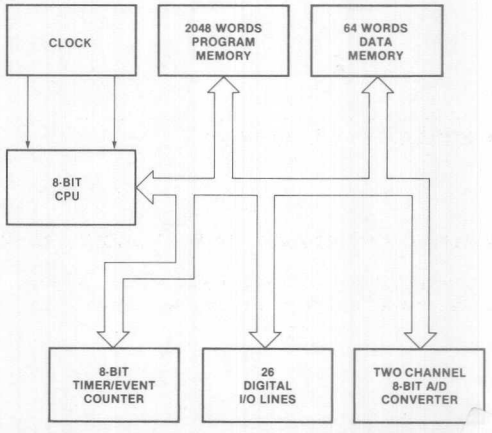
PIN CONFIGURATION



LOGIC SYMBOL



BLOCK DIAGRAM



PIN DESCRIPTION

Designation	Pin #	Function
V_{SS}	20	Circuit GND potential.
V_{CC}	40	+5V circuit power supply.
PROG	37	Output strobe for Intel® 8243 I/O expander.
P00–P07 Port 0	10–17	8-bit open-drain port with comparator inputs. The switching threshold is set externally by V_{TH}. Optional pull-up resistors may be added via ROM mask selection.
V_{TH}	9	Port 0 threshold reference pin.
P10–P17 Port 1	25–32	8-bit quasi-bidirectional port.
P20–P27 Port 2	33–36 38–39 1–2	8-bit quasi-bidirectional port. P20–23 also serve as a 4-bit I/O expander for Intel® 8243.
T0	8	Interrupt input and input pin testable using the conditional transfer instructions JT0 and JNT0. Initiates an interrupt following a low level input if interrupt is enabled. Interrupt is disabled after a reset.
T1	19	Input pin testable using the JT1 and JNT1 conditional transfer instructions. Can be designated the timer/event counter input using the STRT CNT instruction. Also serves as the zero-cross detection input to allow zero-crossover sensing of slowly moving AC inputs. Optional pull-up resistor may be added via ROM mask selection.
RESET	24	Input used to initialize the processor by clearing status flip-flops and setting the program counter to zero.
AV_{SS}	7	A/D converter GND Potential. Also establishes the lower limit of the conversion range.
AV_{CC}	3	A/D +5V power supply.
SUBST	21	Substrate pin used with a bypass capacitor to stabilize the substrate voltage and improve A/D accuracy.
V_{AREF}	4	A/D converter reference voltage. Establishes the upper limit of the conversion range.
AN0, AN1	6,5	Analog inputs to A/D converter. Software selectable on-chip via SEL AN0 and SEL AN1 instructions.
ALE	18	Address Latch Enable. Signal occurring once every 30 input clocks (once every single cycle instruction), used as an output clock.
XTAL 1	22	One side of crystal, inductor, or resistor input for internal oscillator. Also input for external frequency source. (Not TTL compatible.)
XTAL 2	23	Other side of timing control element. This pin is not connected when an external frequency source is used.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias 0°C to 70°C
Storage Temperature -65°C to +180°C
Voltage on Any Pin with
Respect to Ground -0.5V to +7V
Power Dissipation 1 Watt

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permenent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{CC} = 5.5V \pm 1V$, $V_{SS} = 0V$

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage (All except XTAL 1, XTAL 2, Port 0)	-0.5		0.8	V	
V_{IL1}	Input Low Voltage (Port 0)	-0.5		$V_{TH} - 0.1$	V	
V_{IH}	Input High Voltage (All except XTAL 1, XTAL 2, RESET, Port 0)	2.0		V_{CC}	V	$V_{CC} = 5.0V \pm 10\%$
V_{IH1}	Input High Voltage (All except XTAL 1, XTAL 2, RESET, Port 0)	2.4		V_{CC}	V	$V_{CC} = 6.0V \pm 0.5V$
V_{IH2}	Input High Voltage (Port 0)	$V_{TH} + 0.1$		V_{CC}	V	
V_{IH3}	Input High Voltage (RESET, XTAL 1)	3.0		V_{CC}	V	
V_{TH}	Port 0 Threshold Reference Voltage	0		$V_{CC}/2$	V	
V_{OL}	Output Low Voltage			0.45	V	$I_{IL} = 1.6$ mA
V_{OL1}	Output Low Voltage (P10, P11)			2.5	V	$I_{OL} = 7$ mA
V_{OH}	Output High Voltage (All unless Open Drain Option—Port 0)	2.4			V	$I_{OH} = 50\ \mu A$
I_{LI}	Input Leakage Current (T1)			± 10	μA	$V_{IN} = V_{CC}$
I_{LO}	Output Leakage Current (Open Drain Option—Port 0)			± 10	μA	$V_{CC} \geq V_{IN} \geq V_{SS} + 0.45V$
I_{CC}	V_{CC} Supply Current			80	mA	
V_{T1}	Zero-Cross Detection Input (T1)	1		3	VACpp	Input through a capacitor

A.C. CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{CC} = 5.5V \pm 1V$, $V_{SS} = 0V$

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{CY}	Cycle Time	10.0	50.0	μs	3 MHz XTAL = 10 μs
t_{LL}	ALE Pulse Width	4.6		μs	$t_{CY} = 10\ \mu s$
Δ_F	Oscillator Frequency Variation—Resistor Mode	-20	+20	%	F = 2.5 MHz, R = 15 kΩ
F_{T1}	Zero-Cross Detection Input Frequency (T1)	0.03	1	kHz	

A.C. TEST CONDITIONS

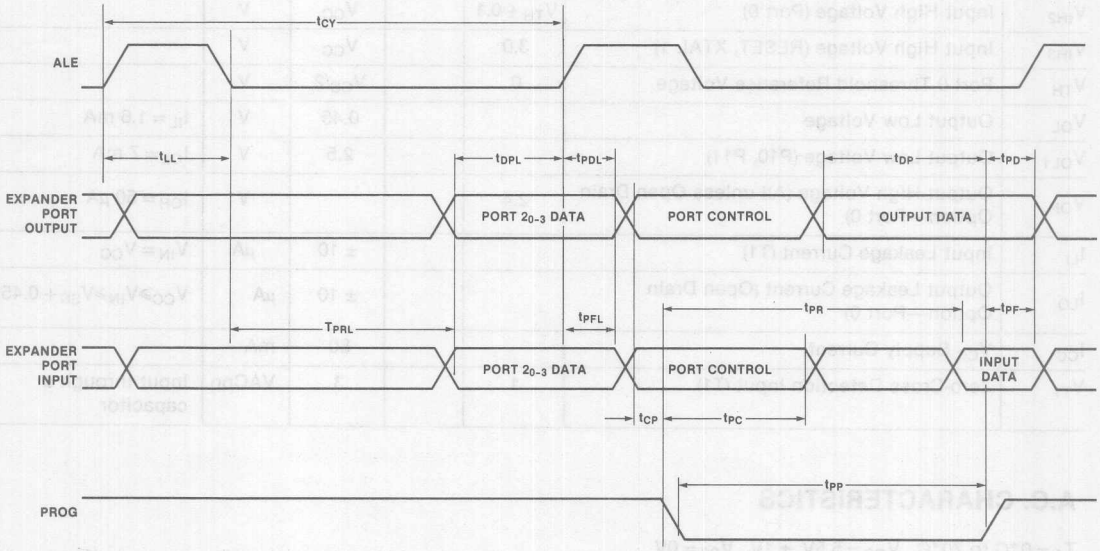
Control Outputs: $C_L = 80$ pF

A.C. CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{CC} = 5.5V \pm 1V$, $V_{SS} = 0V$

	Symbol	Parameter	Min.	Max.	Unit	Notes
Expander Operation	t_{CP}	Port Control Setup Before Falling Edge of PROG	110		ns	
	t_{PC}	Port Control Hold After Falling Edge of PROG	140		ns	
	t_{PR}	PROG to Time P2 Input Must Be Valid		810	ns	
	t_{DP}	Output Data Setup Time	220		ns	
	t_{PD}	Output Data Hold Time	65		ns	
	t_{PF}	Input Data Hold Time	0	150	ns	
	t_{PP}	PROG Pulse Width	1510		ns	
Normal Operation	t_{PRL}	ALE to Time P2 Input Must Be Valid	810		ns	
	t_{DPL}	Output Data Setup Time	400		ns	
	t_{PDL}	Output Data Hold Time	150		ns	
	t_{PFL}	Input Data Hold Time	110		ns	

PORT 2 TIMING

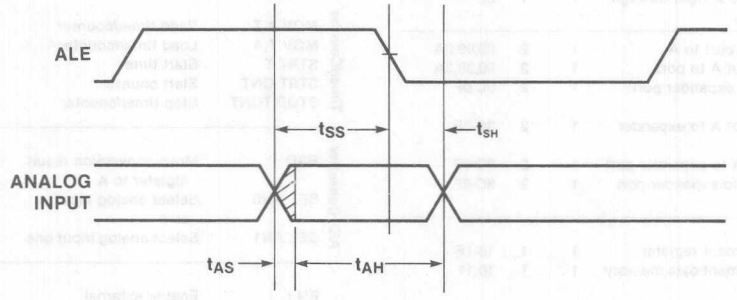


A/D CONVERTER CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{CC} = 5.5V \pm 1V$, $V_{SS} = 0V$, $AV_{CC} = 5.5V \pm 1V$, $AV_{SS} = 0V$

Parameter	Min.	Typ.	Max.	Unit	Comments
Resolution	8			Bits	
Non-Linearity		± ½		LSB	(Note 1)
Zero Error		0		LSB	(Note 2) $T_A = 25°C$
Full Scale Error		0		LSB	(Note 3) $T_A = 25°C$
Quantization Error		± ½		LSB	(Note 4)
Absolute Accuracy		± 1		%	(Note 5)
Conversion Range	AV_{SS}		V_{AREF}	V	
V_{AREF}	$AV_{CC}/2$		AV_{CC}	V	
Input Capacitance (AN0, AN1)		1		pF	
Conversion Time	4		4	t_{CY}	
Sample Hold Time (t_{AS})		0.07		t_{CY}	(Note 6)
Sample Hold Time (t_{AH})		0.23		t_{CY}	(Note 6)
Sample Setup Before Falling Edge of ALE (t_{SS})		0.20		t_{CY}	
Sample Hold After Falling Edge of ALE (t_{SH})		0.10		t_{CY}	

ANALOG INPUT TIMING



NOTES:

1. Non-linearity error is the maximum deviation from a straight line through the end points of the A/D transfer characteristics.
2. Zero error is the difference between the output of an ideal and the actual A/D for zero input voltage.
3. Full-scale error is the difference between the output of an ideal and the actual A/D for full-scale input voltage.
4. Quantization error is the uncertainty caused by the converters finite resolution.
5. Absolute accuracy describes the difference between the actual input voltage and the full-scale weighted equivalent of the binary output. Included are quantizing and all other errors.
6. The analog input must be maintained at a constant voltage during the sampling time (t_{AS}) and the sample hold time (t_{AH}).

INSTRUCTION SET

	Mnemonic	Description	Bytes	Cycle	Hexadecimal Opcode
Accumulator	ADD A,R_r	Add register to A	1	1	68-6F
	ADD A,@R	Add data memory to A	1	1	60-61
	ADD A,#data	Add immediate to A	2	2	03
	ADDC A,R_r	Add register with carry	1	1	78-7F
	ADDC A,@R	Add data memory with carry	1	1	70-71
	ADDC A,#data	Add immediate with carry	2	2	13
	ANL A,R_r	And register to A	1	1	58-5F
	ANL A,@R	And data memory to A	1	1	50-51
	ANL A,#data	And immediate to A	2	2	53
	ORL A,R_r	Or register to A	1	1	48-4F
	ORL A,@R	Or data memory to A	1	1	40-41
	ORL A,#data	Or immediate to A	2	2	43
	XRL A,R_r	Exclusive Or register to A	1	1	D8-DF
	XRL A,@R	Exclusive Or data memory to A	1	1	D0-D1
	XRL A,#data	Exclusive Or immediate to A	2	2	D3
	INC A	Increment A	1	1	17
	DEC A	Decrement A	1	1	07
	CLR A	Clear A	1	1	27
	CPL A	Complement A	1	1	37
	DA A	Decimal adjust A	1	1	57
	SWAP A	Swap nibbles of A	1	1	47
	RL A	Rotate A left	1	1	E7
	RLC A	Rotate A left through carry	1	1	F7
	RR A	Rotate A right	1	1	77
	RRC A	Rotate A right through carry	1	1	67
Input/Output	IN A, P_p	Input port to A	1	2	08,09,0A
	OUT P_pA	Output A to port	1	2	90,39,3A
	MOVD A,P_p	Input expander port to A	1	2	0C-0F
	MOVD P_p,A	Output A to expander port	1	2	3C-3F
	ANLD P_p,A	And A to expander port	1	2	9C-9F
	ORLD P_p,A	Or A to expander port	1	2	8C-8F
Registers	INC R_r	Increment register	1	1	18-1F
	INC @R	Increment data memory	1	1	10-11
Branch	JMP addr	Jump unconditional	2	2	04,24,33,64, 84,A4,C4,E4
	JMPP @A	Jump indirect	1	2	B3
	DJNZ R,addr	Decrement register and jump on R not zero	2	2	E8-EF
	JC addr	Jump on carry = 1	2	2	F6
	JNC addr	Jump on carry = 0	2	2	E6
	JZ addr	Jump on A zero	2	2	C6
	JNZ addr	Jump on A not zero	2	2	96
	JT0	Jump on T0 = 1	2	2	36
	JNT0	Jump on T0 = 0	2	2	26
	JT1 addr	Jump on T1 = 1	2	2	56
	JNT1 addr	Jump on T1 = 0	2	2	46
	JTF addr	Jump on timer flag	2	2	16
Subroutine	CALL	Jump to subroutine	1	2	14,34,54,74, 94,B4,D4,F4
	RET	Return	1	2	83
Flags	CLR C	Clear carry	1	1k	97
	CPL C	Complement carry	1	1	A7
Data Moves	MOV A,R_r	Move register to A	1	1	F8-FF
	MOV A,@R	Move data memory to A	1	1	F0-F1
	MOV A,#data	Move immediate to A	2	2	23
	MOV R_r,A	Move A to register	1	1	A8-AF
	MOV @R,A	Move A to data memory	1	1	A0-A1
	MOV R_r,#data	Move immediate to register	2	2	B8-BF
	MOV @R,#data	Move immediate to data memory	2	2	B0-B1
	XCH A,R_r	Exchange A and register	1	1	28-2F
	XCH A,@R	Exchange A and data memory	1	1	20-21
	XCHD A,@R	Exchange nibble of A and register	1	1	30-'31
	MOVP A,@A	Move to A from current page	1	2	A3
Timer/Counter	MOV A,T	Read timer/counter	1	1	42
	MOV T,A	Load timer/counter	1	1	62
	STRT T	Start timer	1	1	55
	STRT CNT	Start counter	1	1	45
	STOP TCNT	Stop timer/counter	1	1	65
A/D Converter	RAD	Move conversion result register to A	1	2	80
	SEL AN0	Select analog input zero	1	1	85
	SEL AN1	Select analog input one	1	1	95
Interrupts	EN I	Enable external interrupt	1	1	05
	DIS I	Disable external interrupt	1	1	15
	EN TCNTI	Enable timer/counter interrupt	1	1	2 5
	DIS TCNTI	Disable timer/counter interrupt	1	1	35
	RET I	Return from interrupt	1	2	93
	NOP	No operation	1	1	00

SYMBOLS AND ABBREVIATIONS USED

A	Accumulator
addr	11-Bit Program Memory Address
AN0, AN1	Analog Input 0, Analog Input 1
CNT	Event Counter
data	8-Bit Number or Expression
I	Interrupt
P	Mnemonic for "in-page" Operation
P_p	Port Designator (P = 1, 2 or 4-7)
R_r	Register Designator (r = 0-7)
T	Timer
T0, T1	Test 0, Test 1
#	Immediate Data Prefix
@	Indirect Address Prefix

FUNCTIONAL DESCRIPTION

PROGRAM MEMORY

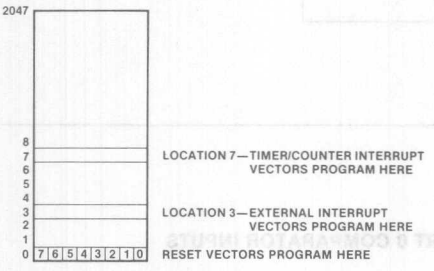
The 8022 program memory consists of 2048 words 8 bits wide which are addressed by the program counter. The memory is ROM which is mask programmable at the factory. No external ROM expansion capability is provided. There are three locations in program memory of special importance.

Location 0: Activating the RESET line of the processor causes the first instruction to be fetched from location 0.

Location 3: Activating the interrupt input line of the processor (if interrupt is enabled) causes a jump to subroutine.

Location 7: A timer/event counter interrupt resulting from a timer/counter overflow causes a jump to subroutine (if timer/counter interrupt is enabled).

Therefore, the first instruction to be executed after initialization is stored in location 0, the first word of an external interrupt service routine is stored in location 3, and the first word of a timer/event counter interrupt service routine is stored in location 7.



PROGRAM MEMORY MAP

Program memory can be used to store constants as well as program instructions. The MOVP instruction allows easy table lookup for constants and display formatting.

DATA MEMORY

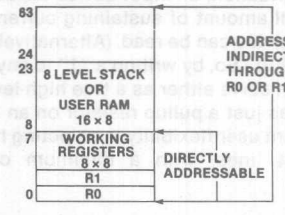
On-chip data memory is organized as 64 words eight bits wide. All locations are indirectly addressable and eight designated locations are directly addressable. Also included in the data memory is the program counter stack, addressed by a 3-bit stack pointer.

The first eight locations (0–7) of the array are designated as working registers and are directly addressable by any of the 11 direct register instructions. These locations are readily accessible for a variety of operations with a minimum number of instruction bytes required for their manipulation. Thus, they are usually used to store frequently accessed intermediate results. The DJNZ instruction makes very efficient use of the working registers as program loop counters by allowing the programmer to decrement and test the register in a single instruction.

Registers 0 and 1 have yet another function in that they can be used to indirectly address all locations in the data memory using the indirect register instructions. These two RAM pointer registers are especially useful for repetitive type operations on adjacent memory locations. The indirect register instruction specifies which pointer register to use and the content of the pointer register is used to address a location in RAM. The contents of the addressed location are used during the execution of the instruction and may be modified. The pointer registers may also point to registers 0–7, if desired.

Locations 8–23 serve a dual role in that they contain the 8-level program counter stack, two RAM locations per level. The program counter stack enables the processor to keep track of the return addresses generated by interrupts or CALL instructions by storing the contents of the program counter prior to servicing the subroutine. A 3-bit stack pointer determines which of the program counter stack's eight register pairs will be loaded with the next return address generated. The stack pointer, when initialized to 000 by RESET, points to RAM locations 8 and 9. The first subroutine jump or interrupt results in the program counter contents being transferred to locations 8 and 9. The stack pointer is then incremented by one and points to locations 10 and 11 in anticipation of another CALL. The end of a subroutine, which is signaled by a return instruction (RET or RETI), causes the stack pointer to be decremented and the contents of the resulting register pair to be transferred to the program counter.

Since the program counter's addresses are 11 bits long, two bytes or registers must be used to store a single address. Thus, the 16-byte program counter stack permits up to a total of 8 levels of subroutine nesting without overflowing the stack. If overflow does occur, the deepest address stored (locations 8 and 9) will be overwritten and lost since the stack pointer overflows from 111 to 000. It also underflows from 000 to 111. If a particular application does not require 8 levels of nesting, tthe unused portion of the program counter stack may be used as any other indirectly addressable RAM location. For example, if only 3 levels of subroutine nesting are used, then only locations 8–13 need be reserved for the program counter stack, and locations 14–23 can be used for data storage.



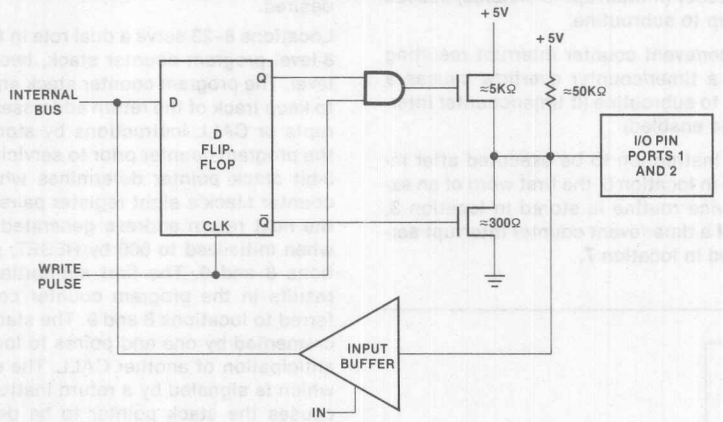
DATA MEMORY MAP

INPUT/OUTPUT

The 8022 has 26 lines which can be used for digital input or output functions. These lines are organized as 3 ports of 8 lines, each of which serve as either inputs, outputs, or bidirectional ports, and 2 test inputs which can alter program sequences when tested by conditional jump instructions.

Ports 1 and 2 have identical operating characteristics and are both quasi-bidirectional. That is, each line may serve as an input, an output, or both. Data written to these ports is statically latched and remains unchanged until rewritten. As inputs, these lines are non-latching; i.e., inputs must be present until read by an input instruction. Inputs are fully TTL compatible and all outputs will drive at least one standard TTL load. Two lines of port 1 (P10 and P11) are designated as high current drive lines and have the ability to sink 7 mA. In addition, these pins may be paralleled for 14 mA output if the output logic states are always the same. The high current output lines eliminate the need for discrete transistors in many applications.

The lines of ports 1 and 2 are quasi-bidirectional because of their output structure which allows them to be used as inputs, outputs, or both, even though as outputs they are statically latched.

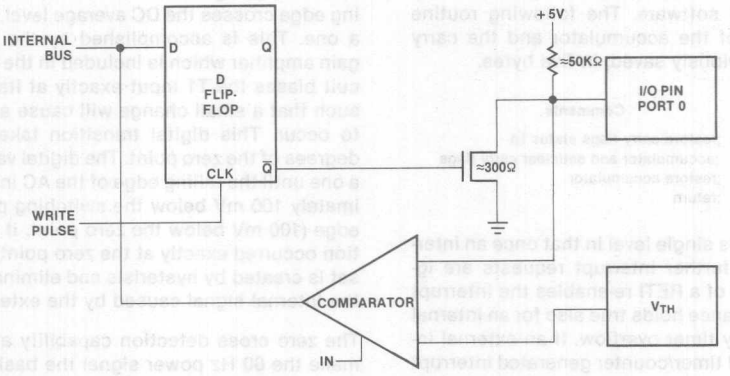


QUASI-BIDIRECTIONAL PORT STRUCTURE

Each line is continuously pulled up to +5V through a relatively high impedance device (~50 kΩ). This pullup is sufficient to provide the source current for a TTL high level, yet can be pulled low by a standard TTL gate, thus allowing the same pin to be used both as an input and output. When writing a "0" or low value to these ports, a low impedance device (~300Ω) overcomes the high pullup and provides TTL current sinking capability. When writing a "1", a large current is momentarily supplied through a relatively low impedance device (~5kΩ) to allow a fast data transfer. After a short time (less than one instruction cycle) the low impedance device is shut off and the small pullup maintains the "1" level indefinitely. In this situation, an input device capable of overriding the small amount of sustaining current supplied by the pullup device can be read. (Alternatively, the data written cn be read.) So, by writing a "1" to any particular pin that pin can serve either as a true high-level latched output pin, or as just a pullup resistor on an input. This allows maximum user flexibility in selecting his input or latched output lines, with a minimum of external components.

PORT 0 COMPARATOR INPUTS

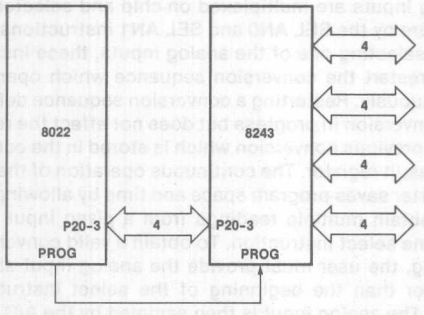
Port 0 has been modified from the standard quasi-bidirectional structure to allow an optional open drain configuration with comparator inputs. The low impedance pullup device has been eliminated and the high impedance pullup is optional. Thus, the user can choose via a mask programmable selection each line of port 0 to be either quasi-bidirectional with a high impedance or true open-drain. The open drain configuration allows the line to sink current through the low impedance pulldown device or to float in the high output state. More importantly, the open drain configuration makes port 0 very easy to drive when it is used as inputs. The input circuitry for each line of port 0 includes a voltage comparator which amplifies the voltage difference between the input port line and the port 0 threshold reference pin (V_{TH}). The voltage gain of the comparator is sufficient to sense a 100 mV input differential withn the range V_{SS} to $V_{CC}/2$.



PORT 0 I/O STRUCTURE

If V_{TH} is allowed to float, it will bias itself to the digital switch point of the other ports, and port 0 behaves as a set of normal digital inputs. However, by biasing V_{TH}, the switch point can be both tightly controlled and adjusted. Common uses for this would include high noise margin inputs ($V_{CC}/2$), unusual logic level inputs as from a diode isolated keyboard, analog channel expansion, and direct capacitive touchpanel interface. The comparator action is automatic and the port is read just as any other port.

In addition to the 26 digital I/O lines contained on-board the 8022, a user can obtain additional I/O lines by utilizing the Intel® 8243 I/O expander chip or standard TTL. The 8243 interfaces to 4 port lines of the 8022 (lower half of port 2) and is strobed by the PROG line of the 8022.



I/O EXPANDER INTERFACE

The 8243 contains four 4-bit I/O ports which serve as extensions of the on-chip I/O and are addressed as ports 4–7. The following operations may be performed on these ports:

1. Transfer Accumulator to Port
2. Transfer Port to Accumulator
3. And Accumulator to Port
4. Or Accumulator to Port

A 4-bit transfer from a port to the lower half of the accumulator sets the most significant four bits to zero. Each transfer consists of two 4-bit nibbles. The first contains the "opcodes" and port address, and the second contains the actual 4 bits of data. A high-to-low transition of the PROG line indicates that address is present while a low-to-high transition indicates the presence of data.

TEST AND INTERRUPT INPUTS

In addition to the 24 general purpose I/O lines which comprise ports 0, 1, and 2, the 8022 has two inputs which are testable via conditional jump instructions, T0 and T1. These pins allow inputs to cause program branches without the necessity to load an input port into the accumulator. T0 and T1 have other functions as well.

The Test 0 pin serves as an external interrupt input as well as a testable input. An interrupt sequence is initiated by applying a low "0" level input to the T0 pin when external interrupt is enabled. Interrupt is level triggered and active low to allow "WIRE ORING" of several interrupt sources at the input pin. When an interrupt is detected, it causes a "jump to subroutine" at location 3 in program memory as soon as all other cycles of the current instruction are complete. At this time, the program counter contents are saved in the program counter stack, but the remaining status of the processor is not. Unlike the 8048, the 8022 does not contain a program status word. Thus, when appropriate, the carry and auxiliary carry flags are saved in software, as the accumulator is. The routine shown below saves the accumulator and the carry flags in only four bytes.

Instructions	Bytes	Comments
MOV R6,A	1	;save accumulator
CLR A	1	;clear accumulator
DA A	1	;convert carry flags into sixes
MOV R7,A	1	;save status of carry flags

The end of an interrupt service subroutine is marked by the execution of a Return from Interrupt instruction (RETI). Prior to returning from the interrupt subroutine

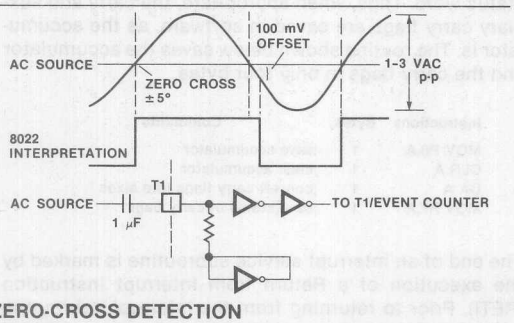
however, the status of the accumulator and the carry flags are restored in software. The following routine restores the status of the accumulator and the carry flags, which was previously saved, in five bytes.

Instructions	Bytes	Comments
MOV A,R7	1	;restore carry flags status to
Add A,#0AAH	2	;accumulator and set/clear carry flags
MOV A,R6	1	;restore accumulator
RETI	1	;return

The interrupt system is single level in that once an interrupt is detected, all further interrupt requests are ignored until execution of a RETI re-enables the interrupt input logic. This sequence holds true also for an internal interrupt generated by timer overflow. If an external interrupt and an internal timer/counter generated interrupt are detected at the same time, the external source will be recognized. If needed, a second external interrupt can be created by enabling the timer/counter interrupt, loading FFH in the counter (one less than terminal count) and enabling the event counter mode. A low-to-high transition on the T1 input will then cause an interrupt vector to location 7.

The Test 1 pin, in addition to being a testable input, serves two other important functions. It can be used as an input pin to the external event counter, as previously mentioned, and it can be used to detect the zero crossing point of slow moving AC signals. Execution of the STRT CNT instruction puts the T1 pin in the counter input mode by connecting T1 to the counter and enabling the counter. Subsequent low-to-high transitions on T1 will cause the counter to increment. Note that this operation differs from the rest of the MCS-48 devices, which increment the counter on high-to-low transitions. This change was made on the 8022 to take advantage of the accuracy of the rising edge detection on the zero cross circuitry. The maximum rate at which the counter may be incremented is once per three instruction cycles (every 30 μs when using a 3 MHz crystal) — there is no minimum frequency.

In addition to serving as a testable input and as the counter input, the T1 pin has special circuitry to detect when an AC signal crosses its average DC level. When driven directly, this pin responds as a normal digital input. To utilize the zero cross detection mode, an AC signal of approximately 1-3 VAC p-p magnitude and a maximum frequency of 1 kHz is coupled through an external capacitor (1 μF) to the T1 pin.



ZERO-CROSS DETECTION

The internal digital state is sensed as a zero until the rising edge crosses the DC average level, when it becomes a one. This is accomplished by the self-biasing high gain amplifier which is included in the T1 input. This circuit biases the T1 input exactly at its switching point, such that a small change will cause a digital transition to occur. This digital transition takes place within 5 degrees of the zero point. The digital value of T1 remains a one until the falling edge of the AC input drops approximately 100 mV below the switching point of the rising edge (100 mV below the zero point, if the digital transition occurred exactly at the zero point). The 100 mV offset is created by hysterisis and eliminates chattering of the internal signal caused by the external noise.

The zero cross detection capability allows the user to make the 60 Hz power signal the basis for this system timing. All timing routines, including time-of-day, can be implemented using the zero cross detection capability of T1 and its conditional jump instructions. In addition, the zero cross detection feature can be used in conjunction with the timer interrupt to interrupt processing at the zero voltage point. This enables the user to control voltage phase sensitive devices such as triacs and SCRs, and to use the 8022 in applications such as shaft speed and angle measurement.

ANALOG TO DIGITAL CONVERTER

The 8022 contains on-chip a complete hardware implementation of an 8-bit analog to digital (A/D) converter with two multiplexed analog inputs. The A/D converter utilizes a successive approximation technique to provide an updated conversion once every four instruction cycles (i.e., once every 40 μs) with a minimum of required software.

The A/D converter consists of four main parts, the input circuitry, a series string of resistors, a voltage comparator, and the successive approximation logic. The two analog inputs are multiplexed on-chip and selected via software by the SEL AN0 and SEL AN1 instructions. Besides selecting one of the analog inputs, these instructions restart the conversion sequence which operates continuously. Restarting a conversion sequence deletes the conversion in progress but does not effect the result of the previous conversion which is stored in the conversion result register. The continuous operation of the A/D converter saves program space and time by allowing the user obtain multiple readings from a given input with only one select instruction. To obtain a valid conversion reading, the user must provide the analog input signal no later than the beginning of the select instruction cycle. The analog input is then sampled by the A/D converter and maintained internally. This voltage becomes one input to the voltage comparator which amplifies the difference between the analog input and the voltage tap on the series resistor string.

The series resistor string is connected between the A/D reference pin (V_{AREF}) and ground (AV_{SS}). It is comprised of 256 identical resistors which divide the voltage between these two pins into 256 identical voltage steps. This configuration gives the converter its inherent monotonicity. The range of V_{AREF} in which full 8-bit resolution can be provided is between $V_{CC}/2$ and V_{CC}.

Thus, the user is given a minimum voltage range from ground to $V_{CC}/2$ and a maximum range from ground to V_{CC} over which 8-bit resolution is insured.

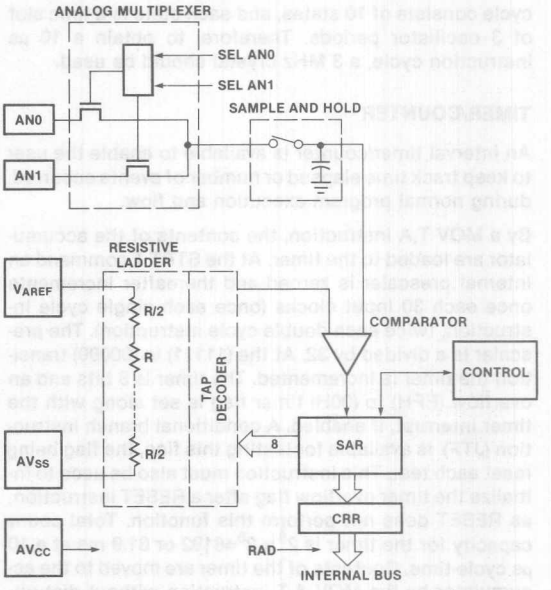
The voltage tap on the series resistor string is selected by the resistor ladder decoder. This decoder is driven by the 8-bit successive approximation register (SAR). Each bit of the SAR is set in succession MSB to LSB and a voltage comparison between the selected resistor ladder voltage and the analog input voltage is performed after the setting of each bit. The result of each comparison determines whether the particular bit will remain set or be reset. All comparisons are performed automatically by the on-chip A/D hardware. At the end of 8 comparisons the SAR contains a valid digital result which is then latched into the conversion result register (CRR). The RAD instruction (read A/D) loads the conversion result from the CRR to the accumulator of the 8022.

As mentioned previously, the software and time required to perform an A/D conversion is optimized by the 8022's on-chip A/D converter configuration. Typical software for reading two sequential A/D conversions and storing them in data memory is shown below:

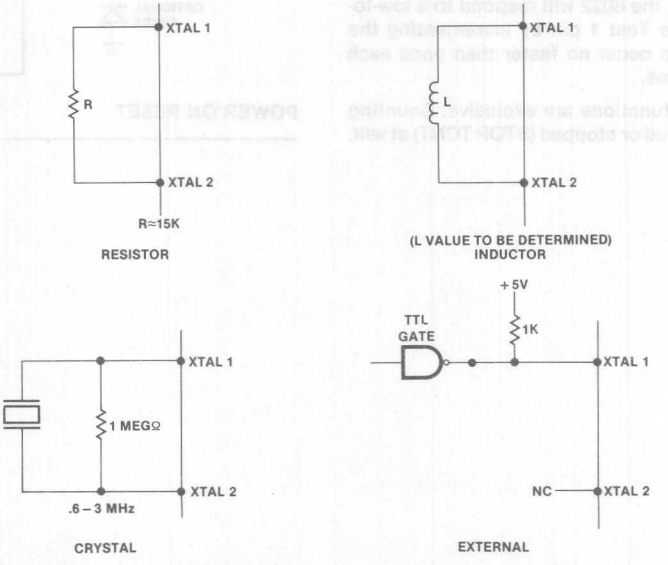
```
First Conversion      SEL AN0        ;Starts conversion of AN0 input
                      MOV R0,#24     ;Set up memory pointer
 50 µs                RAD            ;First conversion value to accumulator
 4 bytes
Second Conversion     MOV @R0,A      ;Store first conversion value
                      INC R0         ;Increment memory location
 40 µs                RAD            ;Second conversion value to accumulator
 3 bytes
```

Note that the second conversion occurs without a second select instruction being used. Rather, the continuous operation of the A/D converter provides an updated digital value 4 instruction cycles after the first.

To insure maximum accuracy from the A/D converter, separate power supply pins (AV_{CC} and AV_{SS}) and a substrate pin (SUBST) have been provided. Supplying the power supply pins with a well filtered and regulated voltage supply minimizes the effect of power supply variance and system noise. The substrate pin should be bypassed to ground through a 500 pF to 0.001 µF capacitor.



A/D CONVERTER BLOCK DIAGRAM



FREQUENCY REFERENCE OPTIONS

OSCILLATOR AND CLOCK

The 8022 contains its own on-board oscillator and clock circuit, requiring only an external timing control element. This control element can be a crystal, inductor, resistor, or clock in. The capacitor normally required in resistor or inductor timing control operation is integrated onto the 8022. All internal time slots are derived from the external element, and all outputs are a function of the oscillator frequency. Pins X1 and X2 are used to input the particular control element. An instruction cycle consists of 10 states, and each state is a time slot of 3 oscillator periods. Therefore, to obtain a 10 μs instruction cycle, a 3 MHz crystal should be used.

TIMER/COUNTER

An interval timer/counter is available to enable the user to keep track time elapsed or number of events occurred, during normal program execution and flow.

By a MOV T,A instruction, the contents of the accumulator are loaded to the timer. At the STRT T command an internal prescaler is zeroed and thereafter increments once each 30 input clocks (once each single cycle instruction, twice each double cycle instruction). The prescaler is a divided by 32. At the (11111) to (00000) transition the timer is incremented. The timer is 8 bits and an overflow (FFH) to (00H) timer flag is set along with the timer interrupt, if enabled. A conditional branch instruction (JTF) is available for testing this flag, the flag being reset each test. This instruction must also be used to initialize the timer overflow flag after a RESET instruction, as RESET does not perform this function. Total count capacity for the timer is $2^8 \times 2^5 = 8192$ or 81.9 ms at a 10 μs cycle time. Contents of the timer are moved to the accumulator by the MOV A,T instruction without disturbing the counting process.

The timer may also be used as an event counter. After a STRT CNT command, the 8022 will respond to a low-to-high transition on the Test 1 pin by incrementing the timer. Transitions can occur no faster than once each three instruction cycles.

The timer and event functions are exclusive. Counting or timing may be started or stopped (STOP TCNT) at will.

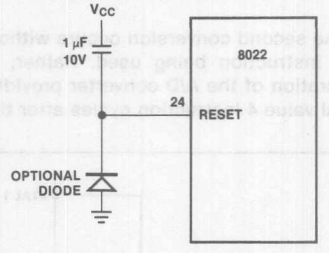
CPU

The 8022 CPU has arithmetic and logical capability. A wide variety of arithmetic and logic instructions may be exercised, which affect the contents of the accumulator, and/or direct or indirect scratchpad locations. Provisions have been made for simplified BCD arithmetic capability through the use of the DAA, SWAP A, and XCHD instructions. In addition, MOVP A,@A allows table lookup for display formatting and constants. Jump conditions such as zero, not zero refer to the accumulator contents at the time of the condition.

RESET

The 8022 may be used in systems with poorly regulated and noisy power supplies. A useful feature is to sense when the power supply dips and quickly recovers, and do a reset to prevent continued operation with incorrect data. This feature may be implemented on the 8022 by connecting a diode between the RESET node and ground.

Including the diode in the reset circuitry forces a reset to occur if the power supply experiences a very sudden voltage glitch. Specifically, if the power supply drops approximately 1.5V and recovers after at least a few nanoseconds, a reset will occur. Without the diode, a power supply interruption of less than 1 ms will not cause a power-on reset.



POWER ON RESET

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

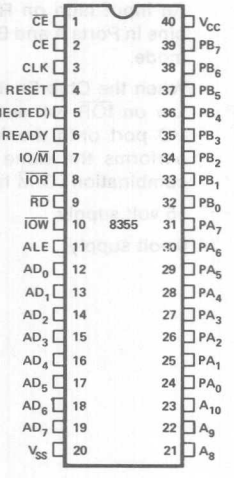
8355
ROM AND I/O EXPANDER

- 2K x 8 ROM
- 2 Eight Bit I/O Ports
- Internal Address Latch
- I/O Lines Individually Assignable as Input or Output
- Single 5V Supply
- 40 Pin DIP
- Completely Interchangeable With 8755 EPROM

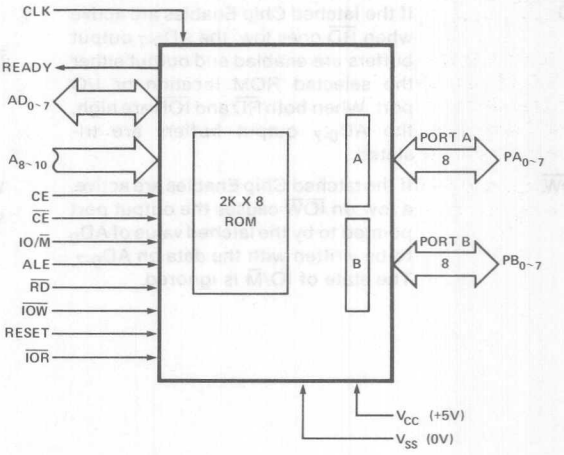
The 8355 is designed to expand both the program memory and I/O capability of the MCS-48™ single component microcomputers (the 8748, 8048 and 8035). This expander increases program memory by 2K words and adds 16 I/O lines to the basic microcomputer without the necessity of any additional components. The completely interchangeable 8755 light erasable EPROM and 8355 mask programmed ROM provide a simple transition from prototype to production. Both versions operate from a single 5V supply and are totally speed compatible with the MCS-48 microcomputers.

The 16 I/O lines are addressed as 2 eight bit I/O ports, yet single lines can be individually designated as input or as output under software control. Outputs are double buffered to prevent any output glitches.

PIN CONFIGURATION



BLOCK DIAGRAM



8355 FUNCTIONAL PIN DEFINITION

Symbol	Function
ALE	When ALE (Address Latch Enable) is high, AD_{0-7}, IO/$\overline{M}$, A_{8-10}, CE, and $\overline{CE}$ enter address latched. The signals (AD, IO/$\overline{M}$, A_{8-10}, CE, $\overline{CE}$) are latched in at the trailing edge of ALE.
AD_{0-7}	Bi-directional Address/Data bus. The lower 8-bits of the ROM or I/O address are applied to the bus lines when ALE is high. During an I/O cycle, Port A or B are selected based on the latched value of AD_0. If $\overline{RD}$ or $\overline{IOR}$ is low when latched Chip Enables are active, the output buffers present data on the bus.
A_{8-10}	These are the high order bits of the ROM address. They do not affect I/O operations.
$\overline{CE}$ CE	Chip Enable Inputs: $\overline{CE}$ is active low and CE is active high. The 8355 can be accessed only when BOTH Chip Enables are active at the time the ALE signal latches them up. If either Chip Enable input is not active, the AD_{0-7} and READY outputs will be in a high impedance state.
IO/$\overline{M}$	If the latched IO/$\overline{M}$ is high when $\overline{RD}$ is low, the output data comes from an I/O port. If it is low the output data comes from the ROM.
$\overline{RD}$	If the latched Chip Enables are active when $\overline{RD}$ goes low, the AD_{0-7} output buffers are enabled and output either the selected ROM location or I/O port. When both $\overline{RD}$ and $\overline{IOR}$ are high, the AD_{0-7} output buffers are tri-stated.
$\overline{IOW}$	If the latched Chip Enables are active, a low on $\overline{IOW}$ causes the output port pointed to by the latched value of AD_0 to be written with the data on AD_{0-7}. The state of IO/$\overline{M}$ is ignored.

Symbol	Function
CLK	The CLK is used to force the READY into its high impedance state after it has been forced low by $\overline{CE}$ low, CE high and ALE high.
READY	Ready is a tri-state output controlled by $\overline{CE}$, CE, ALE and CLK. READY is forced low when the Chip Enables are active during the time ALE is high, and remains low until the rising edge of the next CLK (see Figure 4).
PA_{0-7}	These are general purpose I/O pins. Their input/output direction is determined by the contents of Data Direction Register (DDR). Port A is selected for write operations when the Chip Enables are active and $\overline{IOW}$ is low and a 0 was previously latched from AD_0. Read operation is selected by $\overline{IOR}$ low when the Chip is enabled and AD_0 low. Alternately, IO/$\overline{M}$ high and $\overline{RD}$ low may be used in place of $\overline{IOR}$ when the chip is enabled and AD_0 is low to allow reading from a port.
PB_{0-7}	This general purpose I/O port is identical to Port A except that it is selected by a 1 latched from AD_0.
RESET	An input high on RESET causes all pins in Ports A and B to assume input mode.
$\overline{IOR}$	When the Chip Enables are active, a low on $\overline{IOR}$ will output the selected I/O port onto the AD bus. $\overline{IOR}$ low performs the same function as the combination IO/$\overline{M}$ high and $\overline{RD}$ low.
V_{CC}	+5 volt supply.
V_{SS}	0 volt supply.

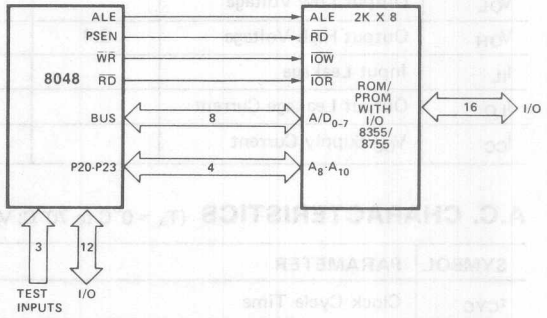
FUNCTIONAL DESCRIPTION

Program Memory — The 8355 contains an 8-bit address latch which allows it to interface directly to MCS-48 Microcomputers without additional hardware. Program memory is accessed by applying 11 bits of address to the A_0 - A_{10} inputs and a low level on the IO/$\overline{M}$ and $\overline{CE}$ inputs then latching these inputs with ALE. The $\overline{CE}$ input serves to select one of several possible 8355s in a system and the IO/$\overline{M}$ signal indicates that a subsequent read operation will be from program memory. While ALE is high the A_0 - A_{10}, IO/$\overline{M}$, and $\overline{CE}$ inputs are allowed into the 8355 and when ALE is brought low, these inputs are latched. If the latched conditions indicate that a program memory fetch is to occur, a low level on $\overline{RD}$ will cause the data to be outputted on the data bus.

I/O Ports — The I/O lines are organized as two 8-bit static ports which can be read or written using the $\overline{IOR}$ and $\overline{IOW}$ control lines. Associated with each port is an 8-bit Data Direction Register (DDR) which serves to define each of the 8 lines of the port as either an input or an output. A "1" bit in the DDR sets the corresponding port bit to the output mode while a "0" designates the input mode. The two least significant bits of the latched address (A_0, A_1) address the two I/O ports and their associated DDR's.

A_1	A_0	Selection
0	0	Port A
0	1	Port B
1	0	DDR A
1	1	DDR B

I/O Port Addressing



Interface to MCS-48™ Microcomputers

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias 0°C to +70°C
Storage Temperature -65°C to +150°C
Voltage on Any Pin
With Respect to Ground -0.3V to +7V
Power Dissipation 1.5W

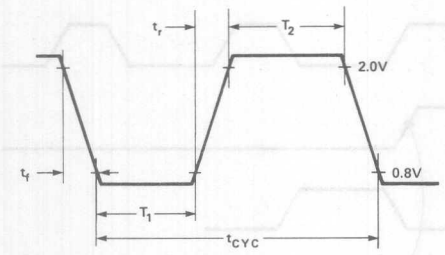
**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

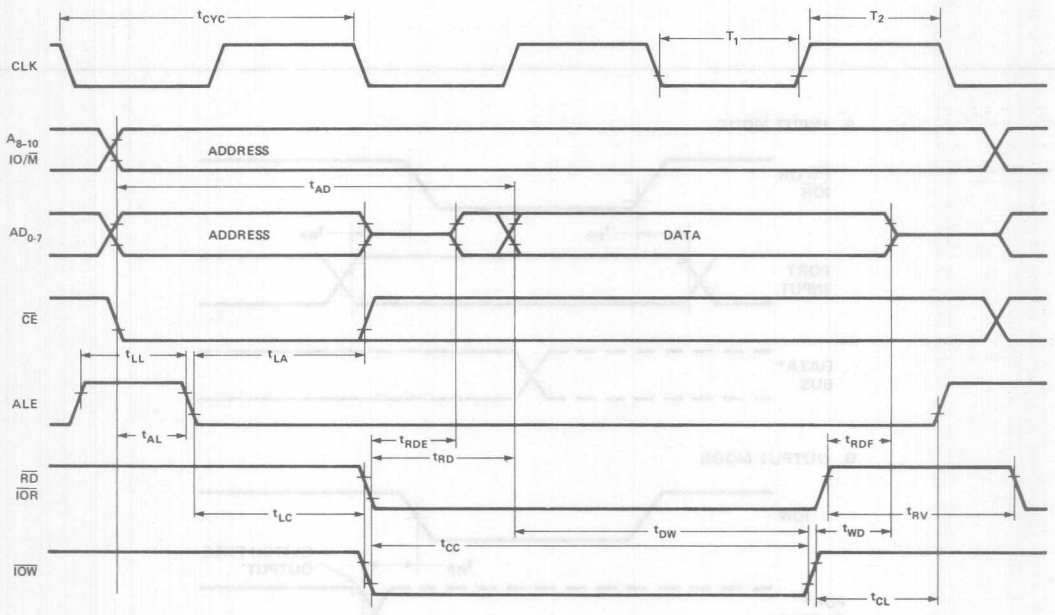
SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5	0.8	V	V_{CC} = 5.0V
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V	V_{CC} = 5.0V
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400μA
I_{IL}	Input Leakage		10	μA	V_{IN} = V_{CC} to 0V
I_{LO}	Output Leakage Current		±10	μA	0.45V ≤V_{OUT} ≤V_{CC}
I_{CC}	V_{CC} Supply Current		180	mA	

A.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

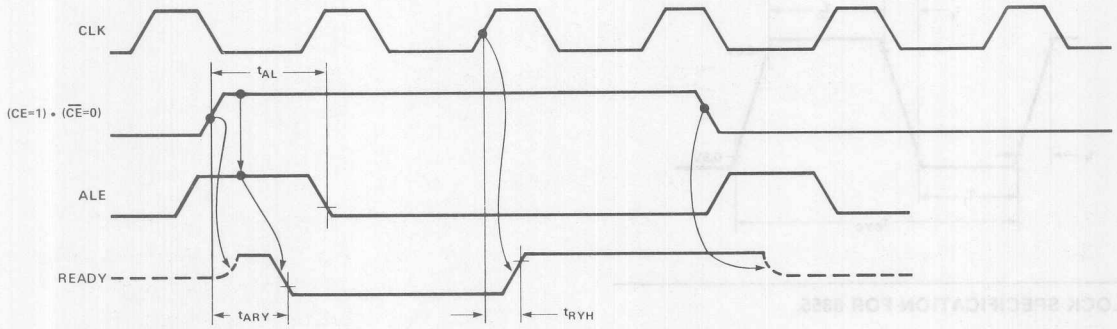
SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
t_{CYC}	Clock Cycle Time	320		ns	C_{LOAD} = 150 pF
T_1	CLK Pulse Width	80		ns	
T_2	CLK Pulse Width	120		ns	
t_f,t_r	CLK Rise and Fall Time		30	ns	
t_{AL}	Address to Latch Set Up Time	50		ns	150 pF Load
t_{LA}	Address Hold Time after Latch	80		ns	
t_{LC}	Latch to READ/WRITE Control	100		ns	
t_{RD}	Valid Data Out Delay from READ Control		170	ns	
t_{AD}	Address Stable to Data Out Valid		400	ns	
t_{LL}	Latch Enable Width	100		ns	
t_{RDF}	Data Bus Float after READ	0	100	ns	
t_{CL}	READ/WRITE Control to Latch Enable	20		ns	
t_{CC}	READ/WRITE Control Width	250		ns	
t_{DW}	Data In to WRITE Set Up Time	150		ns	
t_{WD}	Data In Hold Time After WRITE	10		ns	
t_{WP}	WRITE to Port Output		400	ns	
t_{PR}	Port Input Set Up Time	50		ns	
t_{RP}	Port Input Hold Time	50		ns	
t_{RYH}	READY HOLD TIME	0	160	ns	
t_{ARY}	ADDRESS (CE) to READY		160	ns	
t_{RV}	Recovery Time between Controls	300		ns	
t_{RDE}	Data Out Delay from READ Control	10		ns	



CLOCK SPECIFICATION FOR 8355.

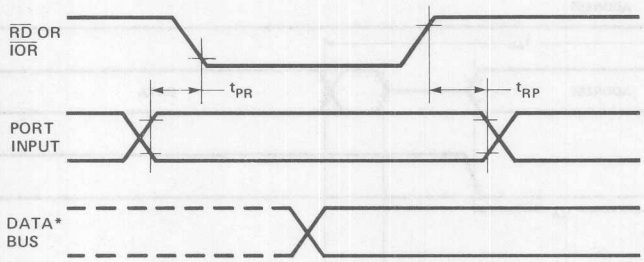


ROM READ AND I/O READ AND WRITE.

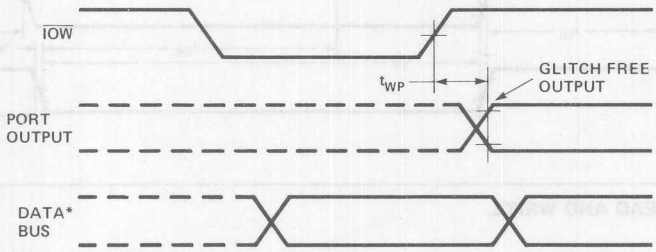


WAIT STATE TIMING (READY = 0).

A. INPUT MODE



B. OUTPUT MODE



*DATA BUS TIMING IS SHOWN IN FIGURE 3.

I/O PORT TIMING.

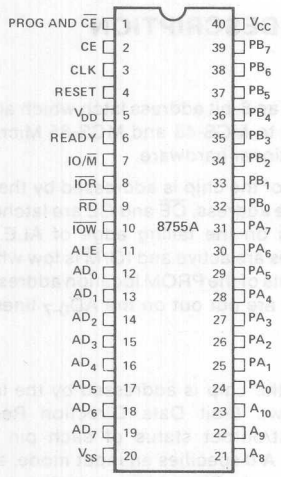
8755A
16,384 BIT EPROM WITH I/O

- 2048 Words x 8 Bits
- Single +5V Power Supply (V_{CC})
- U.V. Erasable and Electrically Reprogrammable
- Internal Address Latch
- 2 General Purpose 8 bit I/O Ports
- Each I/O Port Line Individually Programmable as Input or Output
- Multiplexed Address and Data Bus
- 40 Pin DIP

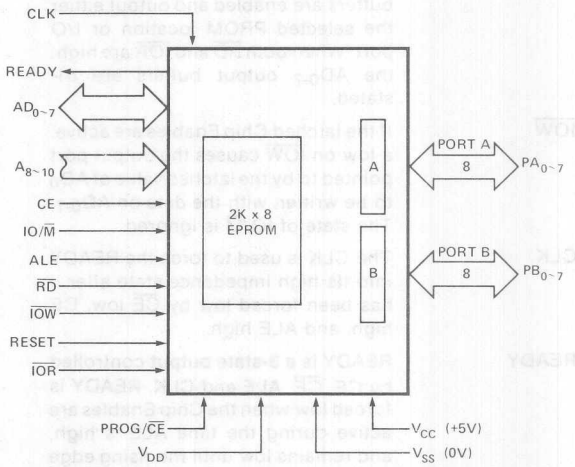
The 8755A is designed to expand both the program memory and I/O capability of the MCS-48™ single component microcomputers (the 8748, 8048 and 8035). This expander increases program memory by 2K words and adds 16 I/O lines to the basic microcomputer without the necessity of any additional components. The completely interchangeable 8755A light erasable EPROM and 8355 mask programmed ROM provide a simple transition from prototype to production. Both versions operate from a single 5V supply and are totally speed compatible with the MCS-48 microcomputers.

The 16 I/O lines are addressed as 2 eight bit I/O ports, yet single lines can be individually designated as input or as output under software control. Outputs are double buffered to prevent any output glitches.

PIN CONFIGURATION



BLOCK DIAGRAM



8755A FUNCTIONAL DESCRIPTION

Symbol	Function
ALE	When Address Latch Enable goes high, AD_{0-7}, IO/M, A_{8-10}, CE, and $\overline{CE}$ enter the address latches. The signals (AD, IO/M, A_{8-10}, CE) are latched in at the trailing edge of ALE.
AD_{0-7}	Bi-directional Address/Data bus. The lower 8-bits of the PROM or I/O address are applied to the bus lines when ALE is high. During an I/O cycle, Port A or B are selected based on the latched value of AD_0. If $\overline{RD}$ or $\overline{IOR}$ is low when the latched Chip Enables are active, the output buffers present data on the bus.
A_{8-10}	These are the high order bits of the PROM address. They do not affect I/O operations.
$\overline{CE}$/PROG	CHIP ENABLE INPUTS: $\overline{CE}$ is active low and CE is active high. Both chip enables must be active to permit accessing the PROM. $\overline{CE}$ is also used as a programming pin (see section on programming).
CE	
IO/$\overline{M}$	If the latched IO/$\overline{M}$ is high when $\overline{RD}$ is low, the output data comes from an I/O port. If it is low the output data comes from the PROM.
$\overline{RD}$	If the latched Chip Enables are active when $\overline{RD}$ goes low, the AD_{0-7} output buffers are enabled and output either the selected PROM location or I/O port. When both $\overline{RD}$ and $\overline{IOR}$ are high, the AD_{0-7} output buffers are tri-stated.
$\overline{IOW}$	If the latched Chip Enables are active, a low on $\overline{IOW}$ causes the output port pointed to by the latched value of AD_0 to be written with the data on AD_{0-7}. The state of IO/$\overline{M}$ is ignored.
CLK	The CLK is used to force the READY into its high impedance state after it has been forced low by $\overline{CE}$ low, CE high, and ALE high.
READY	READY is a 3-state output controlled by CE, $\overline{CE}$, ALE and CLK. READY is forced low when the Chip Enables are active during the time ALE is high, and remains low until the rising edge of the next CLK (see Figure 2.).
PA_{0-7}	These are general purpose I/O pins. Their input/output direction is determined by the contents of Data Direction Register (DDR). Port A is selected for write operations when the Chip Enables are active and $\overline{IOW}$ is low and a 0 was previously latched from AD_0. Read operation is selected by either $\overline{IOR}$ low and active Chip Enables and AD_0 low, or IO/$\overline{M}$ high, $\overline{RD}$ low, active Chip Enables, and AD_0 low.
PB_{0-7}	This general purpose I/O port is identical to Port A except that it is selected by a 1 latched from AD_0.
RESET	In normal operation, an input high on RESET causes all pins in Ports A and B to assume input mode (clear DDR register).
$\overline{IOR}$	When the Chip Enables are active, a low on $\overline{IOR}$ will output the selected I/O port onto the AD bus. $\overline{IOR}$ low performs the same function as the combination of IO/$\overline{M}$ high and $\overline{RD}$ low. When $\overline{IOR}$ is not used in a system, $\overline{IOR}$ should be tied to V_{CC} ("1").
V_{CC}	+5 volt supply.
V_{SS}	Ground Reference.
V_{DD}	**V_{DD} is a programming voltage, and with the 8755A it is tied to +5V when the 8755A is being read.** For programming, a high voltage is supplied with V_{DD}, = 25V, typical.

FUNCTIONAL DESCRIPTION

PROM Section

The 8755A contains an 8-bit address latch which allows it to interface directly to MCS-48 and MCS-85 Microcomputers without additional hardware.

The PROM section of the chip is addressed by the 11-bit address and CE. The address, $\overline{CE}$ and CE are latched into the address latches on the falling edge of ALE. If the latched Chip Enables are active and IO/$\overline{M}$ is low when $\overline{RD}$ goes low, the contents of the PROM location addressed by the latched address are put out on the AD_{0-7} lines.

I/O Section

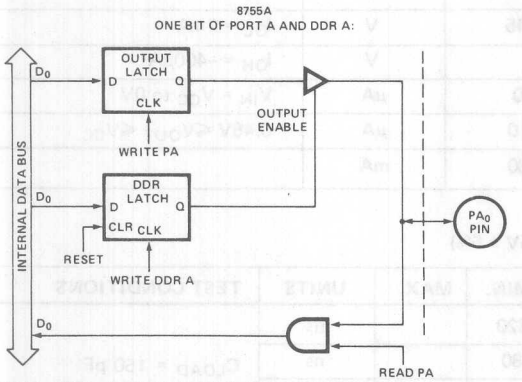
The I/O section of the chip is addressed by the latched value of AD_{0-1}. Two 8-bit Data Direction Registers determine the input/output status of each pin in the corresponding port. A 0 specifies an input mode, and a 1 specifies an output mode. The table summarizes port and DDR designation. Contents of the DDR's cannot be read.

AD_1	AD_0	Selection
0	0	Port A
0	1	Port B
1	0	Port A Data Direction Register (DDR A)
1	1	Port B Data Direction Register (DDR B)

When $\overline{IOW}$ goes low and the Chip Enables are active, the data on the AD is written into I/O port selected by the latched value of AD_{0-1}. During this operation all I/O bits of the selected port are affected, regardless of their I/O mode and the state of IO/$\overline{M}$. The actual output level does not change until $\overline{IOW}$ returns high. (glitch free output).

A port can be read out when the latched Chip Enables are active and either $\overline{RD}$ goes low with IO/$\overline{M}$ high, or $\overline{IOR}$ goes low. Both input and output mode bits of a selected port will appear on lines AD_{0-7}.

To clarify the function of the I/O Ports and Data Direction Registers, the following diagram shows the configuration of one bit of PORT A and DDR A. The same logic applies to PORT B and DDR B.



WRITE PA = ($\overline{IOW}$=0) • (CHIP ENABLES ACTIVE) • (PORT A ADDRESS SELECTED)
WRITE DDR A = ($\overline{IOW}$=0) • (CHIP ENABLES ACTIVE) • (DDR A ADDRESS SELECTED)
READ PA = { [(IO/$\overline{M}$=1) • ($\overline{RD}$=0)] + ($\overline{IOR}$=0) } • (CHIP ENABLES ACTIVE) • (PORT A ADDRESS SELECTED)

Note that hardware RESET or writing a zero to the DDR latch will cause the output latch's output buffer to be disabled, preventing the data in the Output Latch from being passed through to the pin. This is equivalent to putting the port in the input mode. Note also that the data can be written to the Output Latch even though the Output Buffer has been disabled. This enables a port to be initialized with a value prior to enabling the output.

The diagram also shows that the contents of PORT A and PORT B can be read even when the ports are configured as outputs.

ERASURE CHARACTERISTICS

The erasure characteristics of the 8755A are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data show that constant exposure to room level flourescent lighting could erase the typical 8755A in approximately 3 years while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 8755A is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8755 window to prevent unintentional erasure.

The recommended erasure procedure (see page 3-55) for the 8755A is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000μW/cm^2 power rating. The 8755A should be placed within one inch from the lamp tubes during erasure. Some lamps have a filter on their tubes and this filter should be removed before erasure.

PROGRAMMING

Initially, and after each erasure, all bits of the EPROM portions of the 8755A are in the "1" state. Information is introduced by selectively programming "0" into the desired bit locations. A programmed "0" can only be changed to a "1" by UV erasure.

The 8755A can be programmed on the Intel® Universal PROM Programmer (UPP), and the PROMPT™ 80/85 and PROMPT-48™ design aids. The appropriate programming modules and adapters for use in programming both 8755A's and 8755's are shown in the table below.

intel® 8755/8755A Programming Module Cross Reference

MODULE NAME	USE WITH	WILL PROGRAM
UPP 855	UPP[1]	8755
UPP 955	UPP	8755A
UPP UP1[4]	UPP 955	8755
UPP UP2[4]	UPP 855	8755A
PROMPT™ 875	PROMPT 80/85[2]	8755
PROMPT 975	PROMPT 80/85	8755A
PROMPT 475	PROMPT 48[3]	8755A

NOTES:
1. Intel's Universal PROM Programmer module, described on p. 13-45 of the Intel 1977 Data Catalog.
2. Described on p. 13-56 of 1977 Data Catalog.
3. Described on p. 13-51 of 1977 Data Catalog.
4. Special adaptor socket.

PRELIMINARY
Notice: This is not a final specification. Some parameter limits are subject to change.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias -10°C to +70°C
Storage Temperature -65°C to +150°C
Voltage on Any Pin
With Respect to Ground -0.5V to +7V
Power Dissipation 1.5W

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400μA
I_{IL}	Input Leakage		10	μA	V_{IN} = V_{CC} to 0V
I_{LO}	Output Leakage Current		±10	μA	0.45V ≤V_{OUT} ≤V_{CC}
I_{CC}	V_{CC} Supply Current		180	mA	

A.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
t_{CYC}	Clock Cycle Time	320		ns	
T_1	CLK Pulse Width	80		ns	C_{LOAD} = 150 pF (See Figure 3)
T_2	CLK Pulse Width	120		ns	
t_f,t_r	CLK Rise and Fall Time		30	ns	
t_{AL}	Address to Latch Set Up Time	50		ns	
t_{LA}	Address Hold Time after Latch	80		ns	
t_{LC}	Latch to READ/WRITE Control	100		ns	
t_{RD}	Valid Data Out Delay from READ Control		170	ns	
t_{AD}	Address Stable to Data Out Valid		400	ns	150 pF Load
t_{LL}	Latch Enable Width	100		ns	
t_{RDF}	Data Bus Float after READ	0	100	ns	
t_{CL}	READ/WRITE Control to Latch Enable	20		ns	
t_{CC}	READ/WRITE Control Width	250		ns	
t_{DW}	Data In to WRITE Set Up Time	150		ns	
t_{WD}	Data In Hold Time After WRITE	0		ns	
t_{WP}	WRITE to Port Output		400	ns	
t_{PR}	Port Input Set Up Time	50		ns	
t_{RP}	Port Input Hold Time	50		ns	
t_{RYH}	READY HOLD TIME	0	160	ns	
t_{ARY}	ADDRESS (CE) to READY		160	ns	
t_{RV}	Recovery Time between Controls	300		ns	
t_{RDE}	Data Out Delay from READ Control	10		ns	

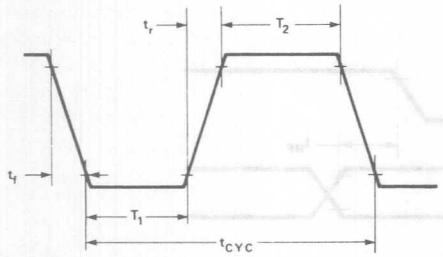


FIGURE 3. CLOCK SPECIFICATION FOR 8755A.

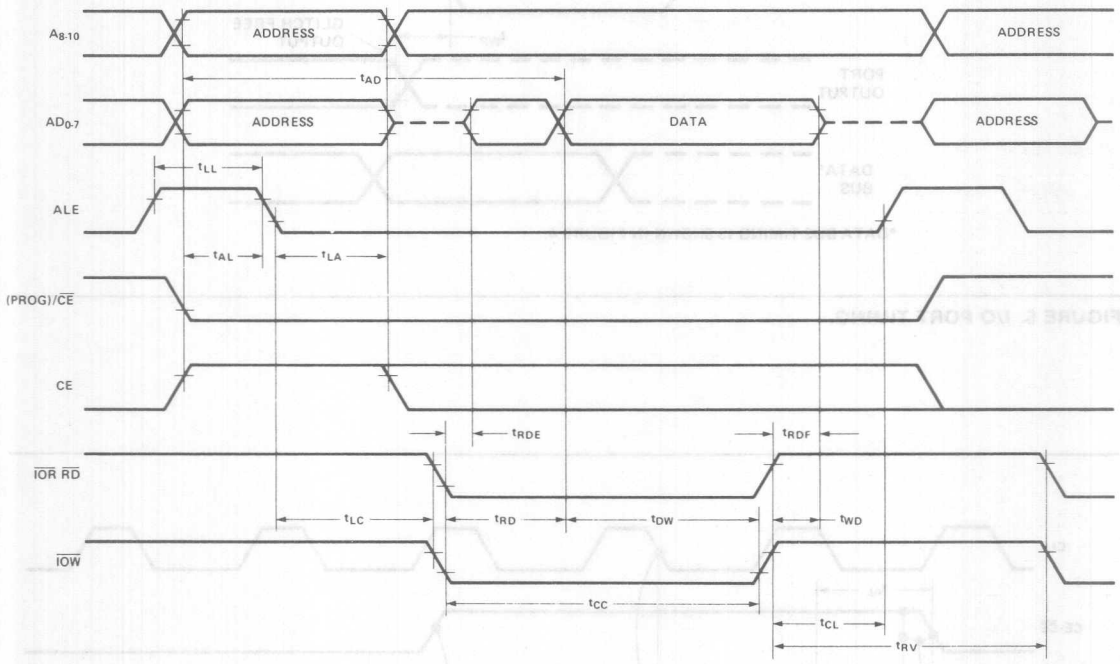


FIGURE 4. PROM READ, I/O READ, AND WRITE TIMING.

Please note that $\overline{CE1}$ must remain low for the entire cycle. This is due to the fact that the programming enable function common to this pin will disrupt internal data bus levels if $\overline{CE1}$ is taken high during the read.

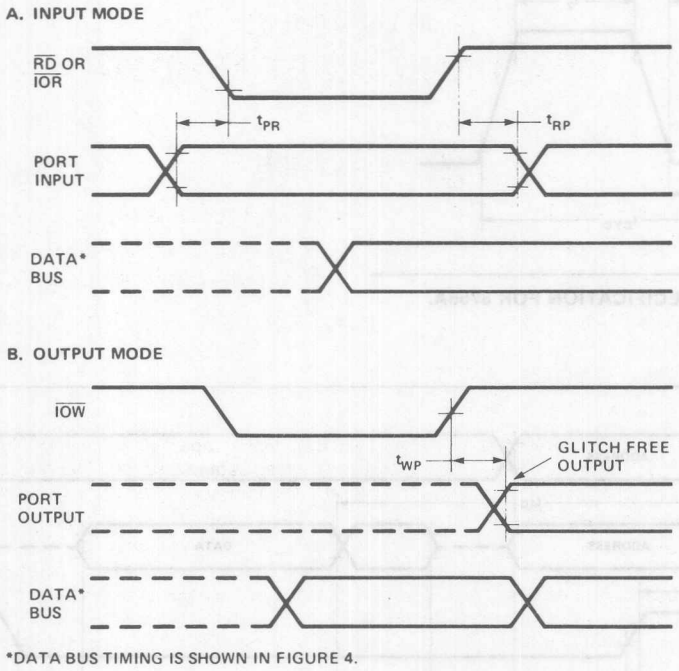


FIGURE 5. I/O PORT TIMING.

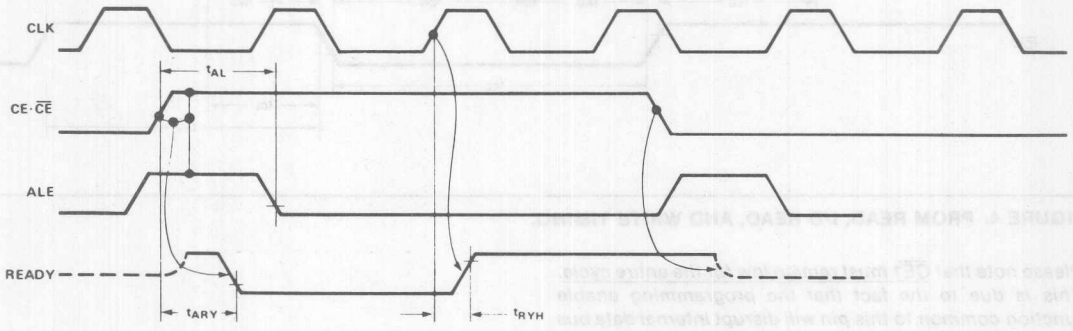


FIGURE 6. WAIT STATE TIMING (READY = 0).

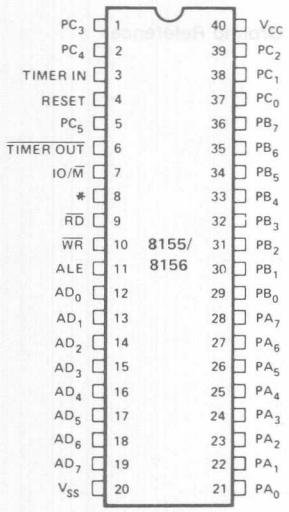
8155/8156
RAM AND I/O EXPANDER

- 256 x 8 Static RAM
- 2 Programmable 8-Bit I/O Ports
- 1 Programmable 6-Bit I/O Port
- Internal Address Latch
- Single 5V Supply
- 40 Pin Dual-In-Line Package
- Programmable 14-Bit Timer/Counter

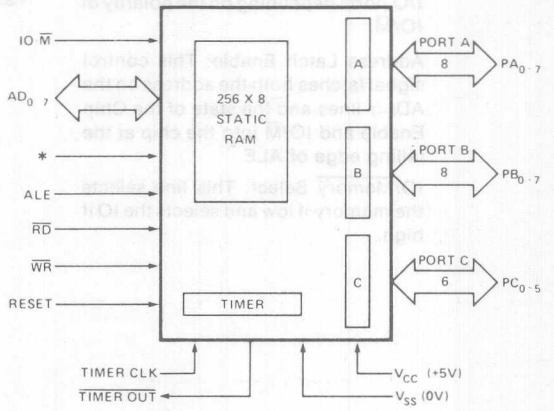
The 8155 is designed to expand the data memory, I/O, and timer capability of the MCS-48™ single component microcomputers (the 8748, 8048, and 8035). This expander increases data memory by 256 words, adds 22 I/O lines, and adds a 14-bit timer/counter to the basic microcomputer without the necessity of any additional components. The 8156 is an 8155 with an active high chip enable (CE) input.

The data memory is a 256 x 8 static RAM which is speed compatible with all MCS-48 components. The I/O consists of two eight-bit ports which can be programmed for either input or output with or without associated handshaking signals and processor interrupt requests. An additional 6-bit port functions as an input port, as an output port, or as the source of strobes for the two eight-bit ports in the handshake mode. The 14-bit programmable timer/counter whose input clock and terminal count output are available to the user externally is programmable for several modes of operation.

PIN CONFIGURATION



BLOCK DIAGRAM



* : 8155 = $\overline{CE}$, 8156 = CE

8155/8156 FUNCTIONAL PIN DEFINITION

The following describes the functions of all of the 8155/8156 pins.

Symbol	Function
RESET	The Reset signal is a pulse provided by the 8085 to initialize the system. Input high on this line resets the chip and initializes the three I/O ports to input mode. The width of RESET pulse should typically be 600 nsec. (Two 8085A clock cycle times).
AD_{0-7}	These are 3-state Address/Data lines that interface with the CPU lower 8-bit Address/Data Bus. The 8-bit address is latched into the address latch on the falling edge of the ALE. The address can be either for the memory section or the I/O section depending on the polarity of the IO/$\overline{M}$ input signal. The 8-bit data is either written into the chip or Read from the chip depending on the status of $\overline{WRITE}$ or $\overline{READ}$ input signal.
CE or $\overline{CE}$	Chip Enable: On the 8155, this pin is $\overline{CE}$ and is ACTIVE LOW. On the 8156, this pin is CE and is ACTIVE HIGH.
$\overline{RD}$	Input low on this line with the Chip Enable active enables the AD_{0-7} buffers. If IO/$\overline{M}$ pin is low, the RAM content will be read out to the AD bus. Otherwise the content of the selected I/O port will be read to the AD bus.
$\overline{WR}$	Input low on this line with the Chip Enable active causes the data on the AD lines to be written to the RAM or I/O ports depending on the polarity of IO/$\overline{M}$.
ALE	Address Latch Enable: This control signal latches both the address on the AD_{0-7} lines and the state of the Chip Enable and IO/$\overline{M}$ into the chip at the falling edge of ALE.
IO/$\overline{M}$	IO/$\overline{Memory}$ Select: This line selects the memory if low and selects the IO if high.

Symbol	Function
$PA_{0-7}(8)$	These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the Command/ Status Register.
$PB_{0-7}(8)$	These 8 pins are general purpose I/O pins. The in/out direction is selected by programming the Command/ Status Register.
$PC_{0-5}(6)$	These 6 pins can function as either input port, output port, or as control signals for PA and PB. Programming is done through the C/S Register. When PC_{0-5} are used as control signals, they will provide the following: PC0 — A INTR (Port A Interrupt) PC1 — A BF (Port A Buffer full) PC2 — $\overline{A\ STB}$ (Port A Strobe) PC3 — B INTR (Port B Interrupt) PC4 — $\overline{B\ BF}$ (Port B Buffer Full) PC5 — $\overline{B\ STB}$ (Port B Strobe)
TIMER IN	This is the input to the counter timer.
$\overline{TIMER\ OUT}$	This pin is the timer output. This output can be either a square wave or a pulse depending on the timer mode.
V_{CC}	+5 volt supply.
V_{SS}	Ground Reference.

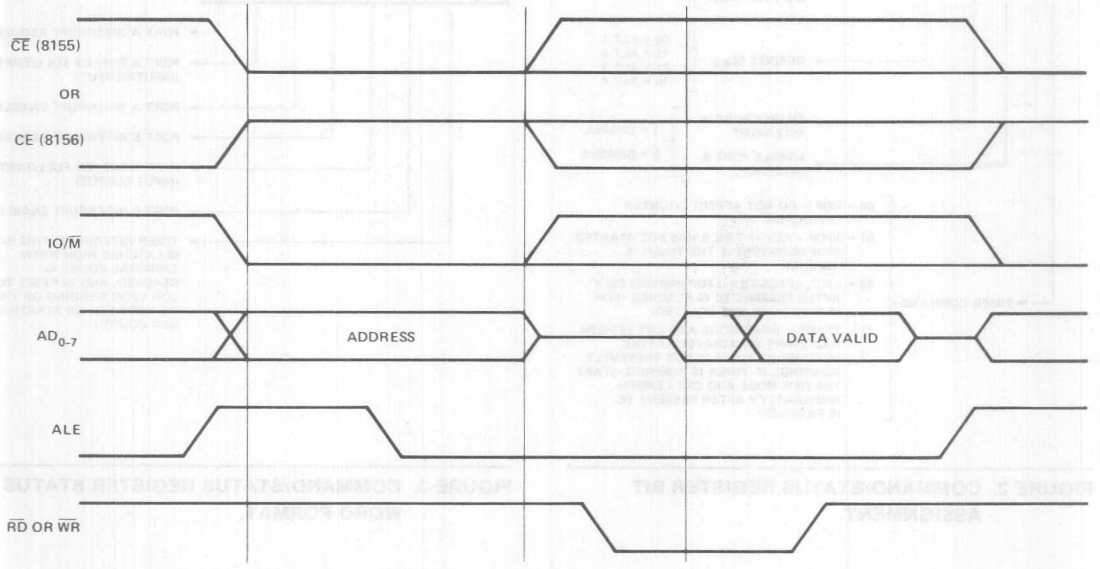
OPERATIONAL DESCRIPTION

The 8155/8156 includes the following operational features:

- 2K Bit Static RAM organized as 256 x 8
- Two 8-bit I/O ports (PA & PB) and one 6-bit I/O port (PC)
- 14-bit binary down counter

The I/O portion contains four registers (Command/Status, PA_{0-7}, PB_{0-7}, PC_{0-5}). The IO/M (IO/Memory Select) pin selects the I/O or the memory (RAM) portion. Detailed descriptions of memory, I/O ports and timer functions will follow.

The 8-bit address on the AD lines, the Chip Enable input, and IO/$\overline{M}$ are all latched on chip at the falling edge of ALE. A low on the IO/$\overline{M}$ must be provided to select the memory section.



NOTE: FOR DETAILED TIMING DIAGRAM INFORMATION, SEE FIGURE 7 AND A.C. CHARACTERISTICS.

FIGURE 1. MEMORY READ/WRITE CYCLE.

PROGRAMMING OF THE COMMAND/ STATUS REGISTER

The command register consists of eight latches one for each bit. Four bits (0-3) define the mode of the ports, two bits (4-5) enable or disable the interrupt from port C when it acts as control port, and the last two bits (6-7) are for the timer.

The C/S register contents can be altered at any time by using the I/O address XXXXX000 during a WRITE operation. The meaning of each bit of the command byte is defined as follows:

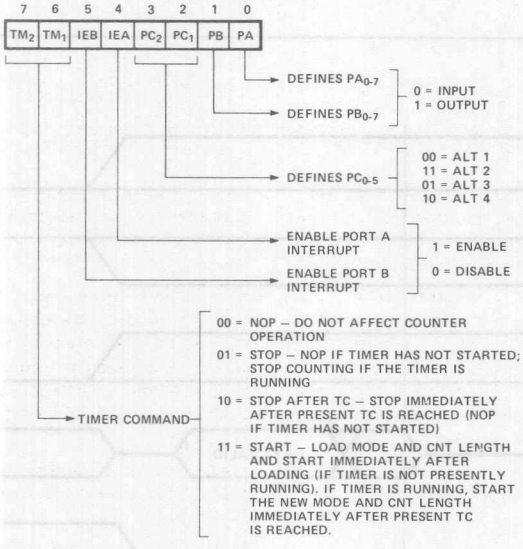


FIGURE 2. COMMAND/STATUS REGISTER BIT ASSIGNMENT.

READING THE COMMAND/STATUS REGISTER

The status register consists of seven latches one for each bit; six (0-5) for the status of the ports and one (6) for the status of the timer.

The status of the timer and the I/O section can be polled by reading the C/S Register (Address XXXXX000). Status word format is shown below:

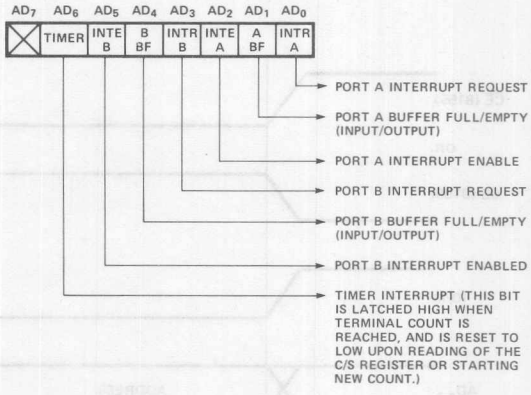


FIGURE 3. COMMAND/STATUS REGISTER STATUS WORD FORMAT.

INPUT/OUTPUT SECTION

The I/O section of the 8155/8156 consists of four registers as described below.

- **Command/Status Register (C/S)** — This register is assigned the address XXXXX000. The C/S address serves the dual purpose.

 When the C/S register is selected during WRITE operation, a command is written into the command register. The contents of this register are *not* accessible through the pins.

 When the C/S (XXXXX000) is selected during a READ operation, the status information of the I/O ports and the timer become available on the AD_{0-7} lines.

- **PA Register** — This register can be programmed to be either input or output ports depending on the status of the contents of the C/S Register. Also depending on the command, this port can operate in either the basic mode or the strobed mode (See timing diagram). The I/O pins assigned in relation to this register are PA_{0-7}. The address of this register is XXXXX001.

- **PB Register** — This register functions the same as PA Register. The I/O pins assigned are PB_{0-7}. The address of this register is XXXXX010.

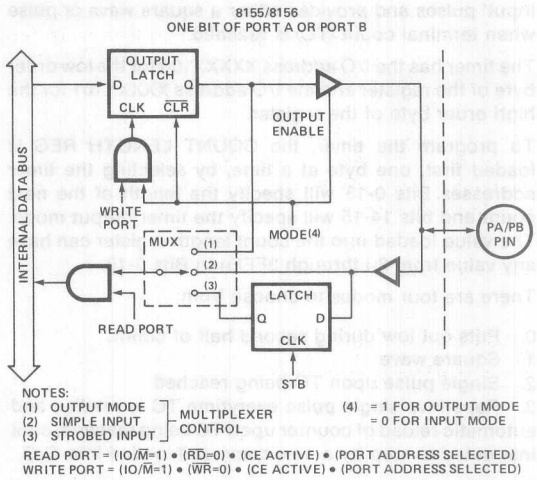
- **PC Register** — This register has the address XXXXX011 and contains only 6-bits. The 6-bits can be programmed to be either input ports, output ports or as control signals for PA and PB by properly programming the AD_2 and AD_3 bits of the C/S register.

 When PC_{0-5} is used as a control port, 3-bits are assigned for Port A and 3 for Port B. The first bit is an interrupt that the 8155 sends out. The second is an output signal indicating whether the buffer is full or empty, and the third is an input pin to accept a strobe for the strobed input mode. See Table 1.

When the 'C' port is programmed to either ALT3 or ALT4, the control signals for PA and PB are initialized as follows:

CONTROL	INPUT MODE	OUTPUT MODE
BF	Low	Low
INTR	Low	High
$\overline{STB}$	Input Control	Input Control

The following diagram shows how I/O PORTS A and B a. structured within the 8155 and 8156:



Note in the diagram that when the I/O ports are programmed to be output ports, the contents of the output ports can still be read by a READ operation when appropriately addressed.

Note also that the output latch is cleared when the port enters the input mode. The output latch cannot be loaded by writing to the port if the port is in the input mode. The result is that each time a port mode is changed from input to output, the output pins will go low. When the 8155/56 is RESET, the output latches are all cleared and all 3 ports enter the input mode.

When in the ALT 1 or ALT 2 modes, the bits of PORT C are structured like the diagram above in the simple input or output mode, respectively.

Reading from an input port with nothing connected to the pins will provide unpredictable results.

TABLE 1. TABLE OF PORT CONTROL ASSIGNMENT.

Pin	ALT 1	ALT 2	ALT 3	ALT 4
PC0	Input Port	Output Port	A INTR (Port A Interrupt)	A INTR (Port A Interrupt)
PC1	Input Port	Output Port	A BF (Port A Buffer Full)	A BF (Port A Buffer Full)
PC2	Input Port	Output Port	A $\overline{STB}$ (Port A Strobe)	A $\overline{STB}$ (Port A Strobe)
PC3	Input Port	Output Port	Output Port	B INTR (Port B Interrupt)
PC4	Input Port	Output Port	Output Port	B BF (Port B Buffer Full)
PC5	Input Port	Output Port	Output Port	B $\overline{STB}$ (Port B Strobe)

The set and reset of INTR and BF with respect to $\overline{STB}$, $\overline{WR}$ and $\overline{RD}$ timing is shown in Figure 8.

To summarize, the registers' assignments are:

Address	Pinouts	Functions	No. of Bits
XXXXX000	Internal	Command/Status Register	8
XXXXX001	PA_{0-7}	General Purpose I/O Port	8
XXXXX010	PB_{0-7}	General Purpose I/O Port	8
XXXXX011	PC_{0-5}	General Purpose I/O Port or Control Lines	6

IMER SECTION

The timer is a 14-bit down counter that counts the 'timer input' pulses and provides either a square wave or pulse when terminal count (TC) is reached.

The timer has the I/O address XXXXX100 for the low order byte of the register and the I/O address XXXXX101 for the high order byte of the register.

To program the timer, the COUNT LENGTH REG is loaded first, one byte at a time, by selecting the timer addresses. Bits 0-13 will specify the length of the next count and bits 14-15 will specify the timer output mode. The value loaded into the count length register can have any value from 2_H through $3FFF_H$ in Bits 0-13.

There are four modes to choose from:

0. Puts out low during second half of count.
1. Square wave
2. Single pulse upon TC being reached
3. Repetitive single pulse everytime TC is readied and automatic reload of counter upon TC being reached, until instructed to stop by a new command loaded into C/S.

Bits 6-7 of Command/Status Register Contents are used to start and stop the counter. There are four commands to choose from:

Note: See the further description on Command/Status Register.

C/S7	C/S6	
0	0	NOP — Do not affect counter operation.
0	1	STOP — NOP if timer has not started; stop counting if the timer is running.
1	0	STOP AFTER TC — Stop immediately after present TC is reached (NOP if timer has not started)
1	1	START — Load mode and CNT length and start immediately after loading (if timer is not presently running). If timer is running, start the new mode and CNT length immediately after present TC is reached.

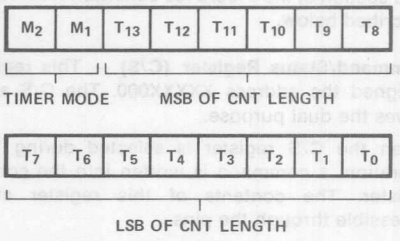
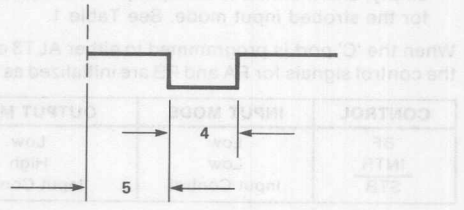


FIGURE 4. TIMER FORMAT

M2 M1 defines the timer mode as follows:

M2	M1	
0	0	Puts out low during second half of count.
0	1	Square wave, i.e., the period of the square wave equals the count length programmed with automatic reload at terminal count.
1	0	Single pulse upon TC being reached.
1	1	Automatic reload, i.e., single pulse everytime TC is reached.

Note: In case of an asymmetric count, i.e. 9, larger half of the count will be high, the larger count will stay active as shown in Figure 5.



Note: 5 and 4 refer to the number of clock cycles in that time period.

FIGURE 5. ASYMMETRIC COUNT.

The counter in the 8155 is not initialized to any particular mode or count when hardware RESET occurs, but RESET does stop the counting. Therefore, counting cannot begin following RESET until a START command is issued via the C/S register.

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias 0°C to +70°C
Storage Temperature -65°C to +150°C
Voltage on Any Pin
With Respect to Ground -0.3V to +7V
Power Dissipation 1.5W

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

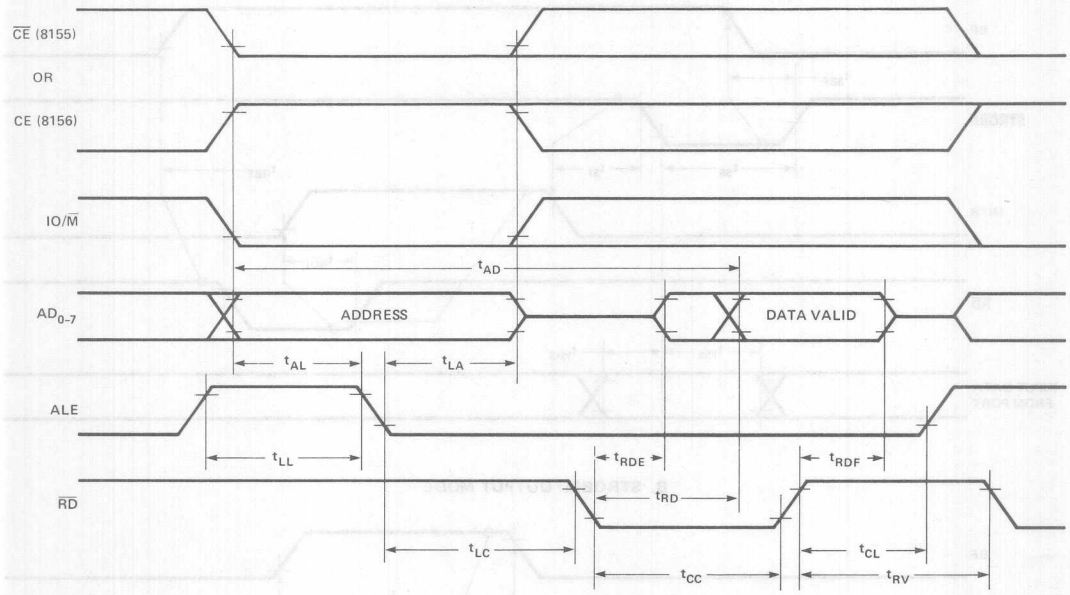
SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}+0.5	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400μA
I_{IL}	Input Leakage		±10	μA	V_{IN} = V_{CC} to 0V
I_{LO}	Output Leakage Current		±10	μA	0.45V ≤V_{OUT} ≤V_{CC}
I_{CC}	V_{CC} Supply Current		180	mA	
I_{IL}(CE)	Chip Enable Leakage 8155 8156		 +100 -100	 μA μA	V_{IN} = V_{CC} to 0V

A.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ± 5%)

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
t_{AL}	Address to Latch Set Up Time	50		ns	
t_{LA}	Address Hold Time after Latch	80		ns	
t_{LC}	Latch to READ/WRITE Control	100		ns	
t_{RD}	Valid Data Out Delay from READ Control		170	ns	
t_{AD}	Address Stable to Data Out Valid		400	ns	
t_{LL}	Latch Enable Width	100		ns	
t_{RDF}	Data Bus Float After READ	0	100	ns	
t_{CL}	READ/WRITE Control to Latch Enable	20		ns	
t_{CC}	READ/WRITE Control Width	250		ns	
t_{DW}	Data In to WRITE Set Up Time	150		ns	
t_{WD}	Data In Hold Time After WRITE	0		ns	
t_{RV}	Recovery Time Between Controls	300		ns	
t_{WP}	WRITE to Port Output		400	ns	
t_{PR}	Port Input Setup Time	70		ns	
t_{RP}	Port Input Hold Time	50		ns	150 pF Load
t_{SBF}	Strobe to Buffer Full		400	ns	
t_{SS}	Strobe Width	200		ns	
t_{RBE}	READ to Buffer Empty		400	ns	
t_{SI}	Strobe to INTR On		400	ns	
t_{RDI}	READ to INTR Off		400	ns	
t_{PSS}	Port Setup Time to Strobe Strobe	50		ns	
t_{PHS}	Port Hold Time After Strobe	120		ns	
t_{SBE}	Strobe to Buffer Empty		400	ns	
t_{WBF}	WRITE to Buffer Full		400	ns	
t_{WI}	WRITE to INTR Off		400	ns	
t_{TL}	TIMER-IN to $\overline{\text{TIMER-OUT}}$ Low		400	ns	
t_{TH}	TIMER-IN to $\overline{\text{TIMER-OUT}}$ High		400	ns	
t_{RDE}	Data Bus Enable from READ Control	10		ns	

Note: For Timer Input Specification, see Figure 10.

A. READ CYCLE



B. WRITE CYCLE

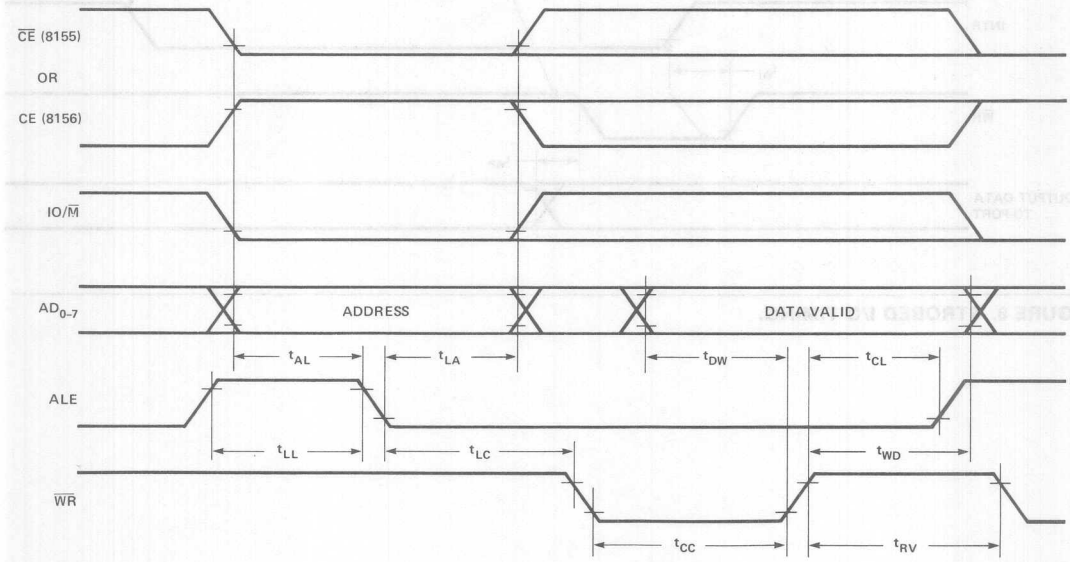


FIGURE 7. 8155/8156 READ/WRITE TIMING DIAGRAMS.

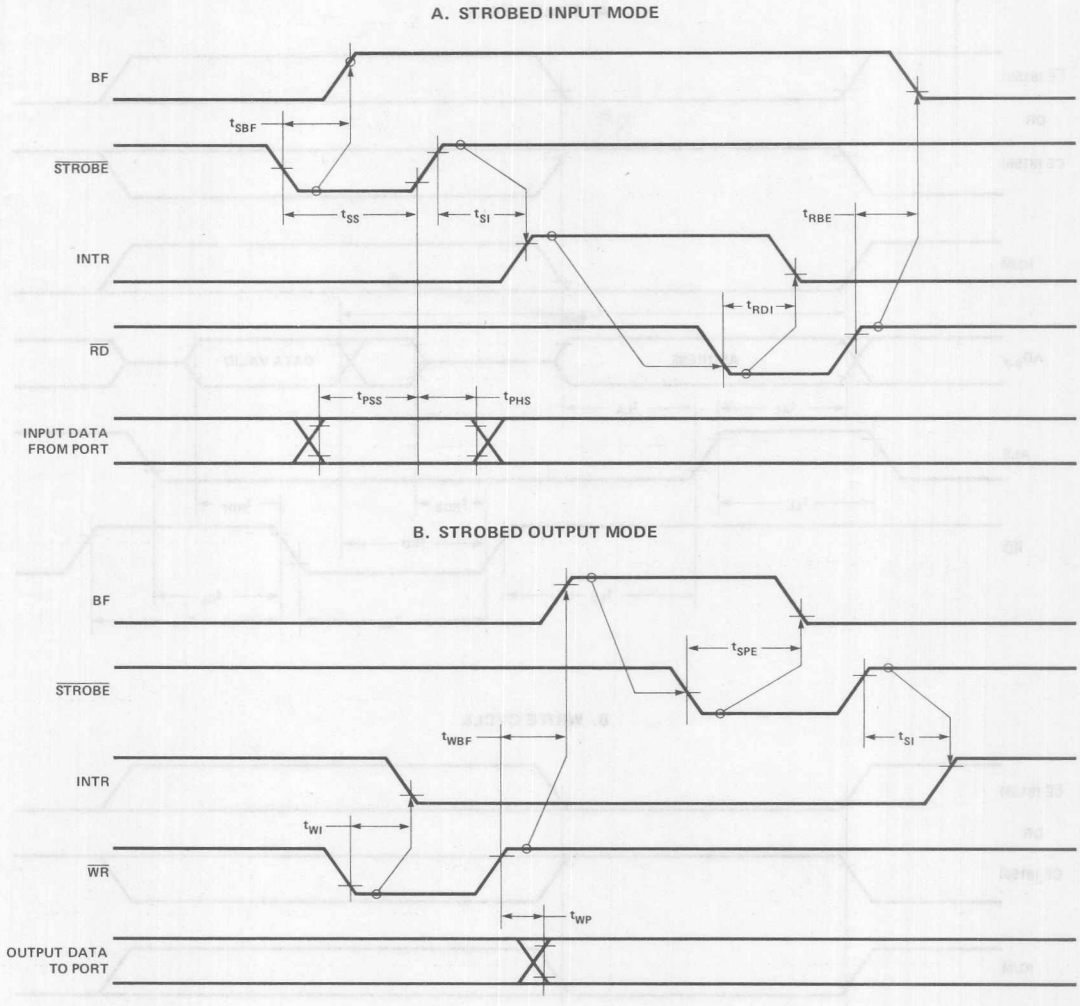


FIGURE 8. STROBED I/O TIMING.

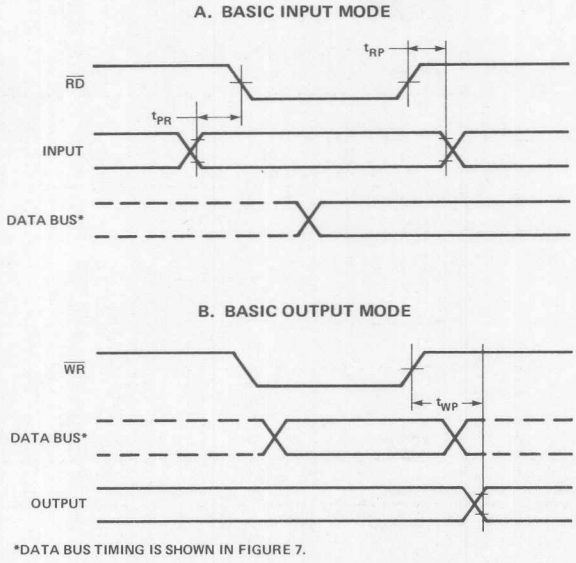
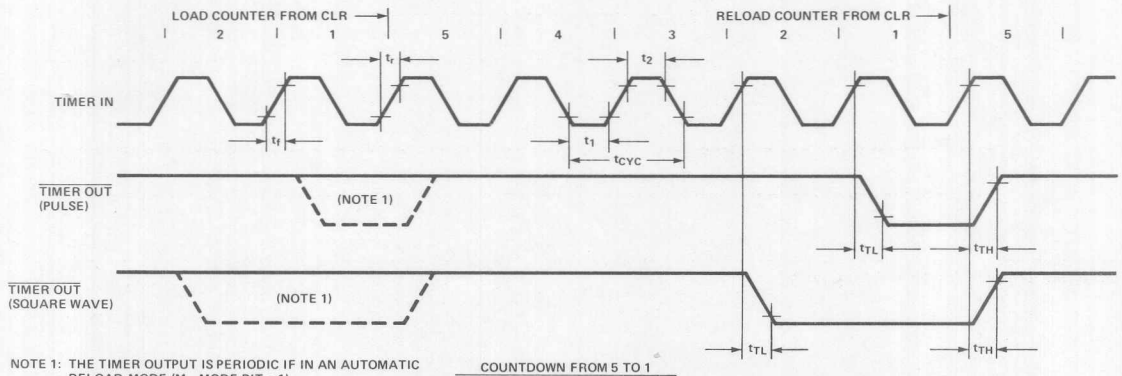


FIGURE 9. BASIC I/O TIMING WAVEFORM.



NOTE 1: THE TIMER OUTPUT IS PERIODIC IF IN AN AUTOMATIC RELOAD MODE (M_1 MODE BIT = 1)

COUNTDOWN FROM 5 TO 1		
t_{CYC}	320 nsec	MIN.
t_r AND t_f	30 nsec	MAX.
t_1	80 nsec	MIN.
t_2	120 nsec	MIN.
t_{TL} AND t_{TH}	400 nsec	MAX.

FIGURE 10. TIMER OUTPUT WAVEFORM.



8243
MCS-48™ INPUT/OUTPUT EXPANDER

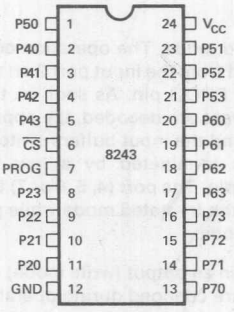
- **Low Cost**
- **Simple Interface to MCS-48™ Microcomputers**
- **Four 4-Bit I/O Ports**
- **AND and OR Directly to Ports**
- **24-Pin DIP**
- **Single 5V Supply**
- **High Output Drive**
- **Direct Extension of Resident 8048 I/O Ports**

The Intel® 8243 is an input/output expander designed specifically to provide a low cost means of I/O expansion for the MCS-48™ family of single chip microcomputers. Fabricated in 5 volts NMOS, the 8243 combines low cost, single supply voltage and high drive current capability.

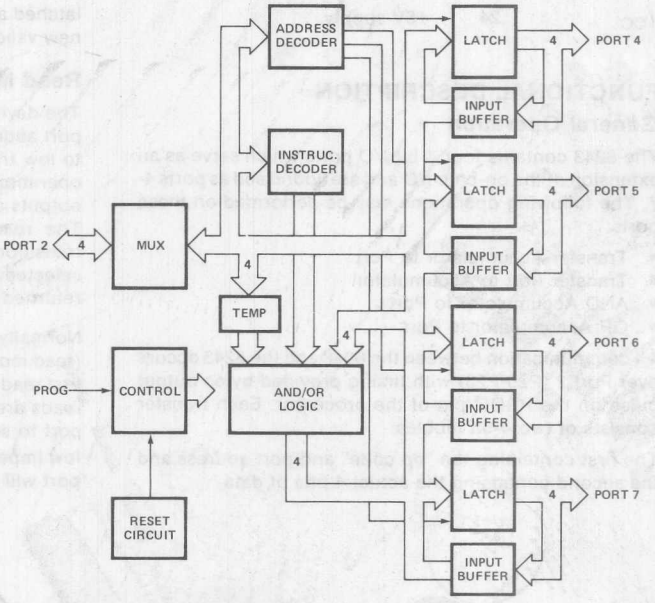
The 8243 consists of four 4-bit bidirectional static I/O ports and one 4-bit port which serves as an interface to the MCS-48 microcomputers. The 4-bit interface requires that only 4 I/O lines of the 8048 be used for I/O expansion, and also allows multiple 8243's to be added to the same bus.

The I/O ports of the 8243 serve as a direct extension of the resident I/O facilities of the MCS-48 microcomputers and are accessed by their own MOV, ANL, and ORL instructions.

PIN CONFIGURATION



BLOCK DIAGRAM



PIN DESCRIPTION

Symbol	Pin No.	Function
PROG	7	Clock Input. A high to low transistion on PROG signifies that address and control are available on P20-P23, and a low to high transition signifies that data is available on P20-23.
$\overline{CS}$	6	Chip Select Input. A high on CS inhibits any change of output or internal status.
P20-P23	11-8	Four-bit bidirectional port contains the address and control bits on a high to low transition of PROG. During a low to high transition contains the data for a selected output port if a write operation, or the data from a selected port before the low to high transition if a read operation.
GND	12	0V supply.
P40-P43 P50-P53 P60-P63 P70-P73	2-5 1,23-21 20-17 13-16	Four-bit bidirectional I/O ports. May be programmed to be input (during read), low impedance latched output (after write) or a tri-state (after read). Data on pins P20-23 may be directly written. ANDed or ORed with previous data.
V_{CC}	24	+5V supply.

FUNCTIONAL DESCRIPTION

General Operation

The 8243 contains four 4-bit I/O ports which serve as an extension of the on-chip I/O and are addressed as ports 4-7. The following operations may be performed on these ports:

- Transfer Accumulator to Port.
- Transfer Port to Accumulator.
- AND Accumulator to Port.
- OR Accumulator to Port.

All communication between the 8048 and the 8243 occurs over Port 2 (P20-P23) with timing provided by an output pulse on the PROG pin of the processor. Each transfer consists of two 4-bit nibbles:

The first containing the "op code" and port address and the second containing the actual 4-bits of data.

A high to low transition of the PROG line indicates that address is present while a low to high transition indicates the presence of data. Additional 8243's may be added to the 4-bit bus and chip selected using additional output lines from the 8048/8748/8035.

Power On Initialization

Initial application of power to the device forces input/output ports 4, 5, 6, and 7 to the tri-state and port 2 to the input mode. The PROG pin may be either high or low when power is applied. The first high to low transition of PROG causes device to exit power on mode. The power on sequence is initiated if V_{CC} drops below 1V.

P21	P20	Address Code	P23	P22	Instruction Code
0	0	Port 4	0	0	Read
0	1	Port 5	0	1	Write
1	0	Port 6	1	0	ORLD
1	1	Port 7	1	1	ANLD

Write Modes

The device has three write modes. MOVD Pi, A directly writes new data into the selected port and old data is lost. ORLD Pi,A takes new data, OR's it with the old data and then writes it to the port. ANLD Pi,A takes new data AND's it with the old data and then writes it to the port. Operation code and port address are latched from the input port 2 on the high to low transition of the PROG pin. On the low to high transition of PROG data on port 2 is transferred to the logic block of the specified output port.

After the logic manipulation is performed, the data is latched and outputed. The old data remains latched until new valid outputs are entered.

Read Mode

The device has one read mode. The operation code and port address are latched from the input port 2 on the high to low transition of the PROG pin. As soon as the read operation and port address are decoded, the appropriate outputs are tri-stated, and the input buffers switched on. The read operation is terminated by a low to high transition of the PROG pin. The port (4, 5, 6 or 7) that was selected is switched to the tri-stated mode while port 2 is returned to the input mode.

Normally, a port will be in an output (write mode) or input (read mode). If modes are changed during operation, the first read following a write should be ignored; all following reads are valid. This is to allow the external driver on the port to settle after the first read instruction removes the low impedance drive from the 8243 output. A read of any port will leave that port in a high impedance state.

Note: The 8243 does not have the same quasi-bidirectional port structure as P1/P2 of the 8048. When a "1" is written to P4-7 of the 8243 it is a "hard 1" (low impedance to +5V) which cannot be pulled low by an external device. All 4 bits of any port can be switched from output mode to input mode by executing a dummy read which leaves the port in a high impedance (no pullup or pulldown) state.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias. 0°C to 70°C
Storage Temperature –65°C to +150°C
Voltage on Any Pin
With Respect to Ground –0.5V to +7V
Power Dissipation 1 Watt

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5V ±10%

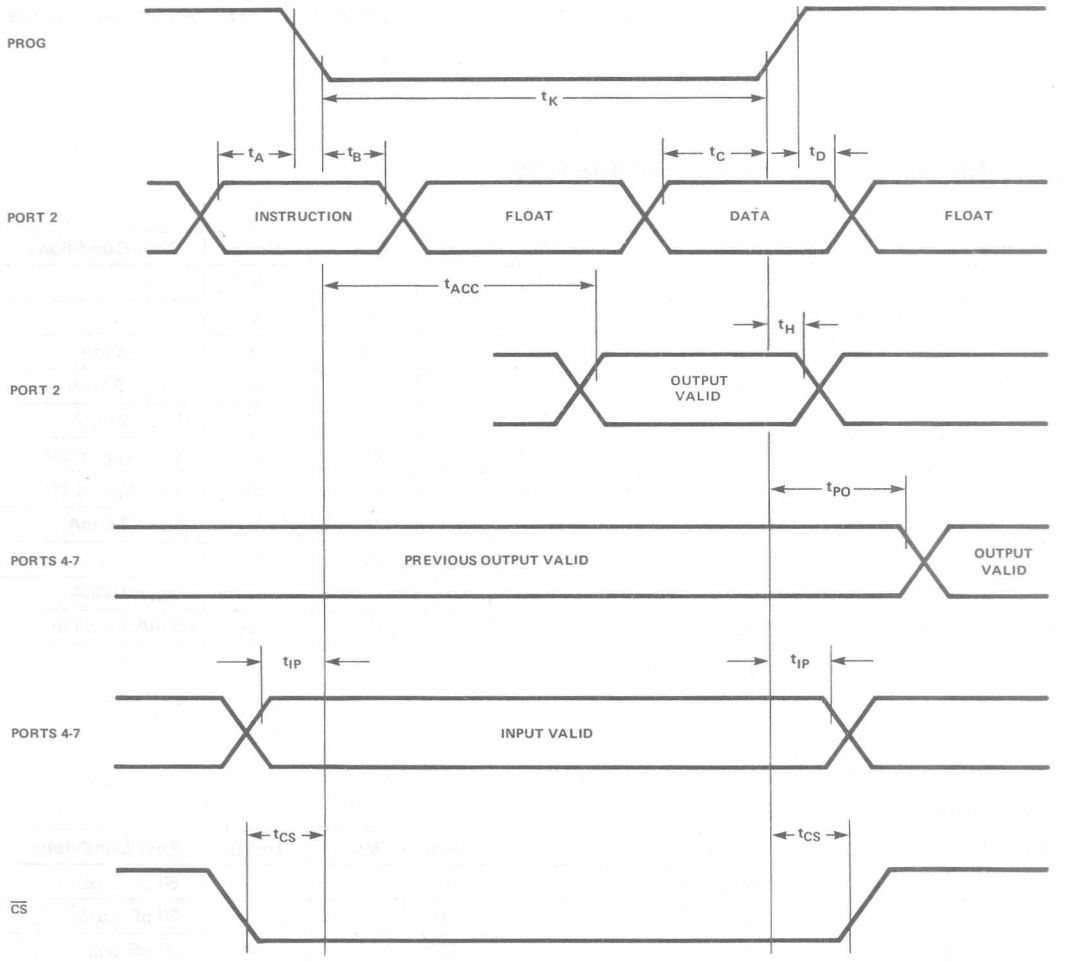
Symbol	Parameter	Min.	Typ.	Max.	Units	Test Conditions
V_{IL}	Input Low Voltage	-0.5		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}+0.5	V	
V_{OL1}	Output Low Voltage Ports 4-7			0.45	V	I_{OL} = 5 mA*
V_{OL2}	Output Low Voltage Port 7			1	V	I_{OL} = 20 mA
V_{OH1}	Output High Voltage Ports 4-7	2.4			V	I_{OH}= 240μA
I_{IL1}	Input Leakage Ports 4-7	-10		20	μA	V_{in} = V_{CC} to 0V
I_{IL2}	Input Leakage Port 2, CS, PROG	-10		10	μA	V_{in} = V_{CC} to 0V
V_{OL3}	Output Low Voltage Port 2			.45	V	I_{OL}= 0.6 mA
I_{CC}	V_{CC} Supply Current		10	20	mA	
V_{OH2}	Output Voltage Port 2	2.4				I_{OH}= 100μA
I_{OL}	Sum of all I_{OL} from 16 Outputs			100	mA	5 mA Each Pin

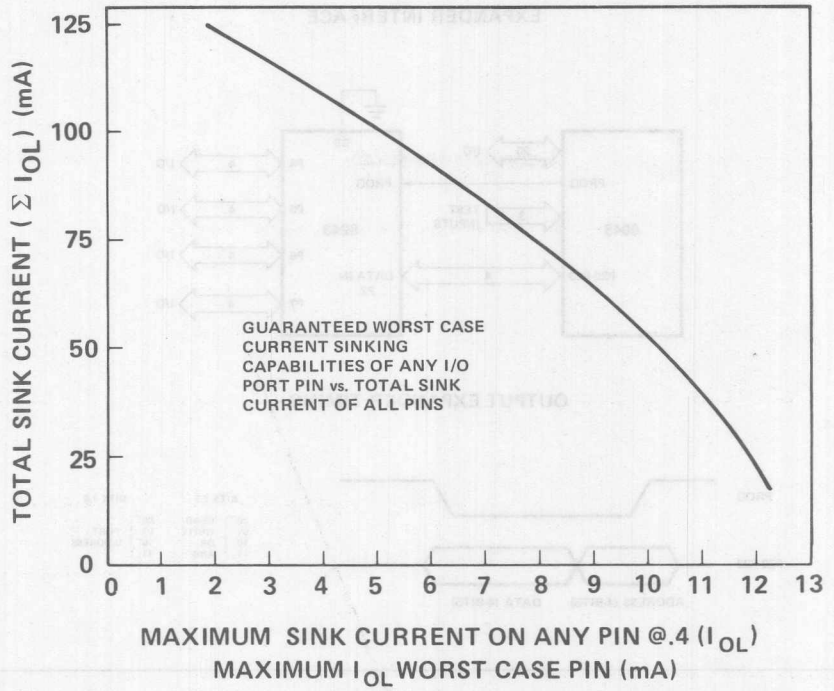
*See following graph for additional sink current capability.

A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5V ± 10%

Symbol	Parameter	Min.	Max.	Units	Test Conditions
t_A	Code Valid Before PROG	100		ns	80 pF Load
t_B	Code Valid After PROG	60		ns	20 pF Load
t_C	Data Valid Before PROG	200		ns	80 pF Load
t_D	Data Valid After PROG	20		ns	20 pF Load
t_H	Floating After PROG	0	150	ns	20 pF Load
t_K	PROG Negative Pulse Width	700		ns	
t_{CS}	CS Valid Before/After PROG	50		ns	
t_{PO}	Ports 4-7 Valid After PROG		700	ns	100 pF Load
t_{LP1}	Ports 4-7 Valid Before/After PROG	100		ns	
t_{ACC}	Port 2 Valid After PROG		650	ns	80 pF Load


PROG
t_K
t_A
t_B
t_C
t_D
PORT 2
INSTRUCTION
FLOAT
DATA
FLOAT
t_{ACC}
t_H
PORT 2
OUTPUT VALID
t_{PO}
PORTS 4-7
PREVIOUS OUTPUT VALID
OUTPUT VALID
t_{IP}
t_{IP}
PORTS 4-7
INPUT VALID
t_{CS}
t_{CS}
$\overline{CS}$



Sink Capability

The 8243 can sink 5 mA @ .4V on each of its 16 I/O lines simultaneously. If, however, all lines are not sinking simultaneously or all lines are not fully loaded, the drive capability of any individual line increases as is shown by the accompanying curve.

For example, if only 5 of the 16 lines are to sink current at one time, the curve shows that each of those 5 lines is capable of sinking 10 mA @ .4V (if any lines are to sink 10 mA the total I_{OL} must not exceed 50 mA or five 10 mA loads).

Example: How many pins can drive 5 TTL loads (1.6 mA) assuming remaining pins are unloaded?

I_{OL} = 5 x 1.6 mA = 8 mA
ϵI_{OL} = 70 mA from curve
pins = 70 mA ÷ 8 mA/pin = 8.75 = 8

In this case, 8 lines can sink 8 mA for a total of 64 mA. This leaves 6 mA sink current capability which can be divided in any way among the remaining 8 I/O lines of the 8243.

Example: This example shows how the use of the 20 mA sink capability of Port 7 affects the sinking capability of the other I/O lines.

An 8243 will drive the following loads simultaneously.

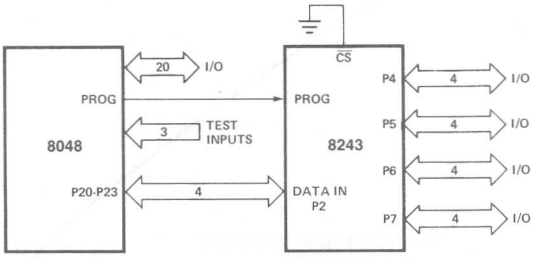
2 loads — 20 mA @ 1V (port 7 only)
8 loads — 5 mA @ .4V
6 loads — 3.2 mA @ .4V
Is this within the specified limits?

ϵI_{OL} = (2 x 20) + (8 x 5) + (6 x 3.2) = 99.2 mA from the curve: for I_{OL} = 5 mA, ϵI_{OL} = 100 mA since 99.2 mA < 100 mA the loads are within specified limits.

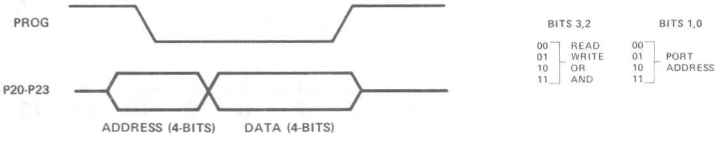
Although the 20 mA @ 1V loads are used in calculating ϵI_{OL}, it is the largest current required @ .4V which determines the maximum allowable ϵI_{OL}.

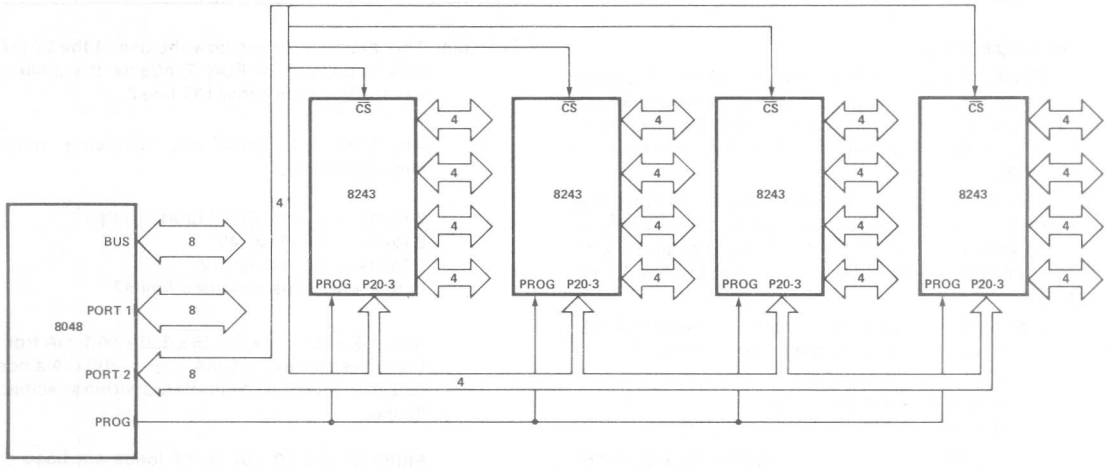
Note: A 10 to 50KΩ pullup resistor to +5V should be added to 8243 outputs when driving to 5V CMOS directly.

EXPANDER INTERFACE



OUTPUT EXPANDER TIMING





USING MULTIPLE 8243's

Chapter 7

COMPATIBLE MCS-80™ COMPONENTS



COMPATIBLE MCS-80™ COMPONENTS

8308	8192 Bit Static MOS ROM	7-1
2316E	16,384 Bit Static MOS ROM	7-5
8708	8192 1K x 8 EPROM	7-9
2716	16K UV Erasable PROM	7-13
8101A-4	1024 Bit Static MOS RAM With Separate I/O	7-17
8111A-4	1024 Bit Static MOS RAM With Common I/O	7-21
5101	1024 Bit Static CMOS RAM	7-25
8212	Eight-Bit Input/Output Port	7-29
8255A	Programmable Peripheral Interface	7-39
8251A	Programmable Communication Interface	7-61
8205	High Speed 1 Out of 8 Binary Decoder	7-77
8214	Priority Interrupt Control Unit	7-83
8216/8226	4-Bit Parallel Bi-Directional Bus Driver	7-87
8253,8253-5	Programmable Interval Timer	7-93
8259	Programmable Interrupt Controller	7-105
8279,8279-5	Programmable Peripheral Interface	7-121
8278	Programmable Keyboard Interface	7-133
8041/8741	Universal Peripheral Interface 8-Bit Microcomputers	7-143

8308
8192 BIT STATIC MOS READ ONLY MEMORY

Organization — 1024 Words x 8 Bits

- **Fast Access — 450 ns**
- **Directly Compatible with 8080 CPU at Maximum Processor Speed**
- **Two Chip Select Inputs for Easy Memory Expansion**
- **Directly TTL Compatible — All Inputs and Outputs**
- **Three State Output — OR-Tie Capability**
- **Fully Decoded**
- **Standard Power Supplies +12V DC, 5V DC**

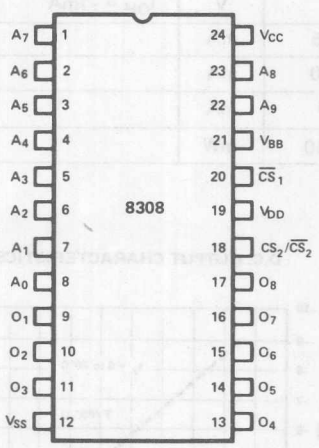
The Intel® 8308 is an 8,192 bit static MOS mask programmable Read Only Memory organized as 1024 words by 8-bits. This ROM is designed for 8080 microcomputer system applications where high performance, large bit storage, and simple interfacing are important design objectives. The inputs and outputs are fully TTL compatible.

A pin for pin compatible electrically programmed erasable ROM, the Intel® 8708, is available for system development and small quantity production use.

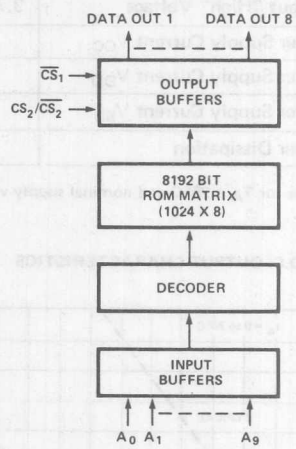
Two Chip Selects are provided — $\overline{CS}_1$ which is negative true, and $CS_2/\overline{CS}_2$ which may be programmed either negative or positive true at the mask level.

The 8308 read only memory is fabricated with N-channel silicon gate technology. This technology provides the designer with high performance, easy-to-use MOS circuits.

PIN CONFIGURATION



BLOCK DIAGRAM



PIN NAMES

A_0- A_9	ADDRESS INPUTS
O_1- O_8	DATA OUTPUTS
$\overline{CS}_1$, CS_2	CHIP SELECT INPUTS

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias -25°C to +85°C
Storage Temperature -65°C to +150°C
Voltage On Any Pin With Respect
To V_{BB} -0.3V to 20V
Power Dissipation 1.0 Watt

*COMMENT

Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

PROGRAMMING

The programming specifications are described in the PROM/ROM Programming Instructions on page 6-74.

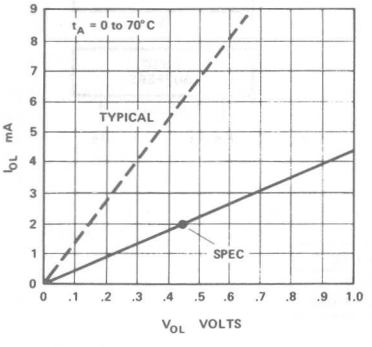
D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5V ±5%; V_{DD} = 12V ±5%, V_{BB} = -5V ±5%, V_{SS} = 0V Unless Otherwise Specified.

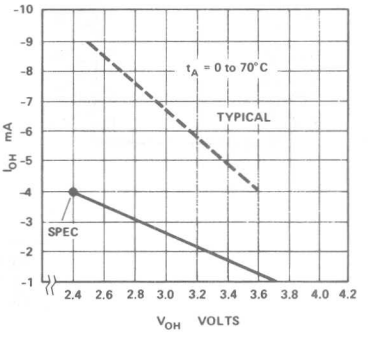
Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.[1]	Max.		
I_{LI}	Input Load Current (All Input Pins Except $\overline{CS}_1$)			±10	μA	V_{IN} = 0 to 5.25V
I_{LCL}	Input Load Current on $\overline{CS}_1$			-1.6	mA	V_{IN} = 0.45V
I_{LPC}	Input Peak Load Current on $\overline{CS}_1$			-4	mA	V_{IN} = 0.8V to 3.3V
I_{LKC}	Input Leakage Current on $\overline{CS}_1$			10	μA	V_{IN} = 3.3V to 5.25V
I_{LO}	Output Leakage Current			10	μA	Chip Deselected
V_{IL}	Input "Low" Voltage	V_{SS}-1		0.8V	V	
V_{IH}	Input "High" Voltage	3.3		V_{CC}+1.0	V	
V_{OL}	Output "Low" Voltage			0.45	V	I_{OL} = 2mA
V_{OH1}	Output "High" Voltage	2.4			V	I_{OH} = -4mA
V_{OH2}	Output "High" Voltage	3.7			V	I_{OH} = -1mA
I_{CC}	Power Supply Current V_{CC}		10	15	mA	
I_{DD}	Power Supply Current V_{DD}		32	60	mA	
I_{BB}	Power Supply Current V_{BB}		10μA	1	mA	
P_D	Power Dissipation		460	840	mW	

NOTE 1: Typical values for T_A = 25°C and nominal supply voltage

D.C. OUTPUT CHARACTERISTICS



D.C. OUTPUT CHARACTERISTICS



A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = +5V ±5%; V_{DD} = +12V ±5%, V_{BB} = -5V ±5%, V_{SS} = 0V, Unless Otherwise Specified.

Symbol	Parameter	Limits[2]			Unit
		Min.	Typ.	Max.	
t_{ACC}	Address to Output Delay Time		200	450	ns
t_{CO1}	Chip Select 1 to Output Delay Time		85	160	ns
t_{CO2}	Chip Select 2 to Output Delay Time		125	220	ns
t_{DF}	Chip Deselect to Output Data Float Time		125	220	ns

NOTE 2: Refer to conditions of Test for A.C. Characteristics. Add 50 nanoseconds (worst case) to specified values at V_{OH} = 3.7V @ I_{OH} = -1mA, C_L = 100pF.

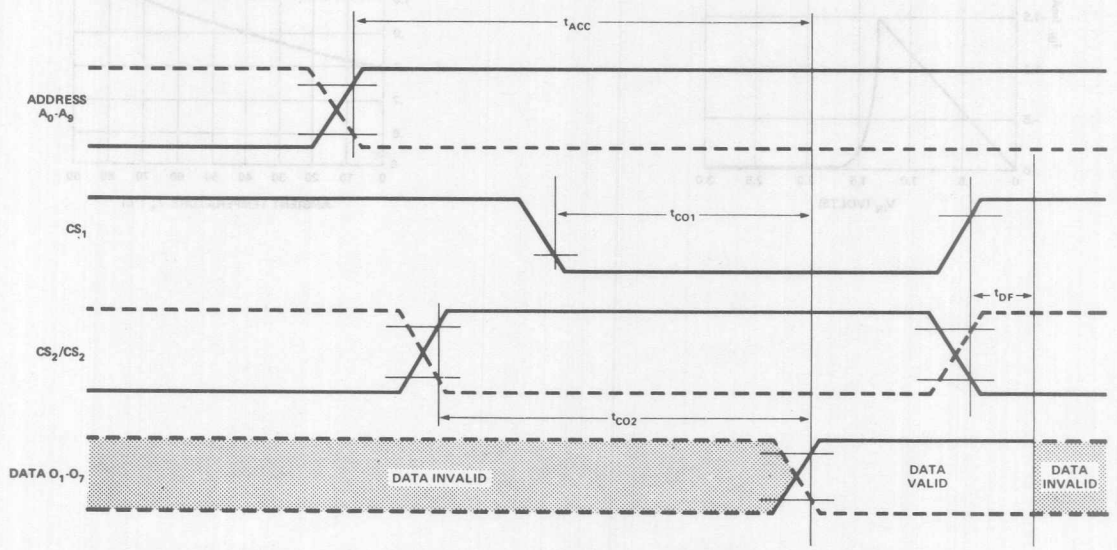
CONDITIONS OF TEST FOR A.C. CHARACTERISTICS

Output Load 1 TTL Gate, and C_{LOAD} = 100pF
Input Pulse Levels 65V to 3.3V
Input Pulse Rise and Fall Times 20 nsec
Timing Measurement Reference Level
. 2.4V V_{IH}, V_{OH}; 0.8V V_{IL}, V_{OL}

CAPACITANCE

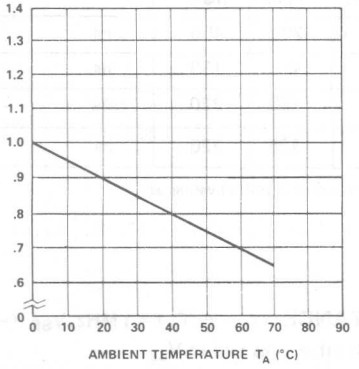
T_A = 25°C, f = 1 MHz, V_{BB} = -5V, V_{DD}, V_{CC} and all other pins tied to V_{SS}.

Symbol	Test	Limits	
		Typ.	Max.
C_{IN}	Input Capacitance		6pF
C_{OUT}	Output Capacitance		12pF

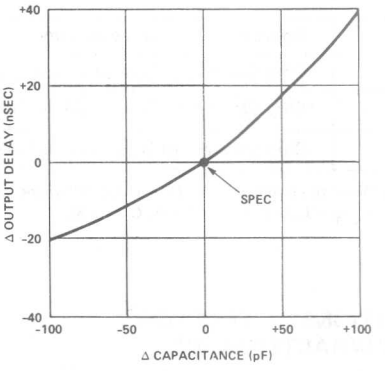


TYPICAL CHARACTERISTICS (Nominal supply voltages unless otherwise noted.)

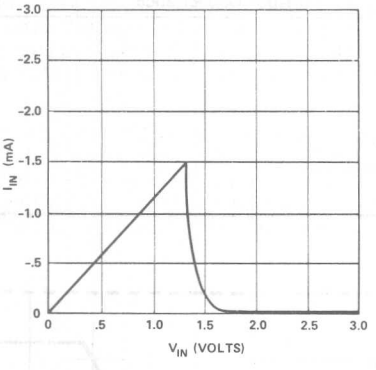
I_{DD} VS. TEMPERATURE (NORMALIZED)



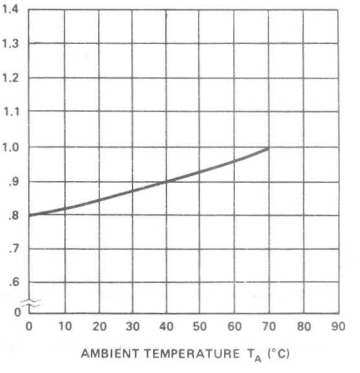
Δ OUTPUT CAPACITANCE VS. Δ OUTPUT DELAY



$\overline{CS}_1$ INPUT CHARACTERISTICS



T_{ACC} VS. TEMPERATURE (NORMALIZED)



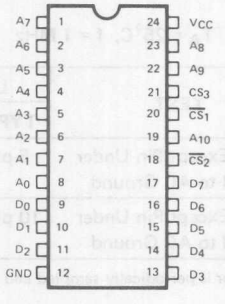
2316E
16K (2K x 8) ROM

- **Fast Access Time—450 ns Max.**
- **Single +5V ± 10% Power Supply**
- **Intel MCS 80 and 85 Compatible**
- **Three Programmable Chip Selects for Simple Memory Expansion and System Interface**
- **EPROM/ROM Pin Compatible for Cost-Effective System Development**
- **Completly Static Operation**
- **Inputs and Outputs TTL Compatible**
- **Three-State Output for Direct Bus Interface**

The Intel® 2316E is a 16,384-bit static, N-channel MOS read only memory (ROM) organized as 2048 words by 8 bits. Its high bit density is ideal for large, non-volatile data storage applications such as program storage. The three-state outputs and TTL input/output levels allow for direct interface with common system bus structures. The 2316E single +5V power supply and 450 ns access time are both ideal for usage with high performance microcomputers such as the Intel MCS™-80 and MCS™-85 devices.

A cost-effective system development program may be implemented by using the pin compatible Intel 2716 16K UV EPROM for prototyping and the lower cost 2616 PROM and 2316E ROM for production. The 2716 is fully compatible to the 2316E in all respects. The three 2316E programmable chip selects may be defined by the user and are fixed during the masking process. To simplify the conversion from 2716 prototyping to 2316E production, it is recommended that the 2316E programmable chip select logic levels be defined the same as that shown in the below data sheet pin configuration. This pin configuration and these chip select logic levels are the same as the 2716.

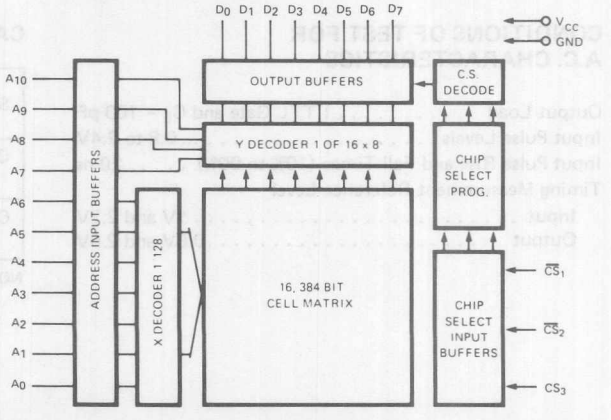
PIN CONFIGURATION



PIN NAMES

A_0–A_{10}	ADDRESS INPUTS
O_7–O_0	DATA OUTPUTS
CS_1–CS_3	CHIP SELECT INPUTS

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias.–10°C to 80°C
Storage Temperature–65°C to +150°C
Voltage On Any Pin With Respect
to Ground –0.5V to +7V
Power Dissipation 1.0 Watt

**COMMENT:* Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = 5V ±10%, unless otherwise specified.

SYMBOL	PARAMETER	LIMITS			UNIT	TEST CONDITIONS
		MIN.	TYP.(1)	MAX.		
I_{LI}	Input Load Current (All Input Pins)			10	μA	V_{IN} = 0 to 5.25V
I_{LOH}	Output Leakage Current			10	μA	Chip Deselected, V_{OUT} = 4.0V
I_{LOL}	Output Leakage Current			–20	μA	Chip Deselected, V_{OUT} = 0.4V
I_{CC}	Power Supply Current		70	120	mA	All Inputs 5.25V Data Out Open
V_{IL}	Input "Low" Voltage	–0.5		0.8	V	
V_{IH}	Input "High" Voltage	2.2		V_{CC}+1.0V	V	
V_{OL}	Output "Low" Voltage			0.4	V	I_{OL} = 2.1 mA
V_{OH}	Output "High" Voltage	2.4			V	I_{OH} = – 400 μA

NOTE: 1. Typical values for T_A = 25°C and nominal supply voltage.

A.C. CHARACTERISTICS

T_A = 0°C to +70°C, V_{CC} = +5V ±10%, unless otherwise specified.

SYMBOL	PARAMETER	LIMITS		UNIT
		MIN.	MAX.	
t_A	Address to Output Delay Time		450	ns
t_{CO}	Chip Select to Output Enable Delay Time		120	ns
t_{DF}	Chip Deselect to Output Data Float Delay Time	10	100	ns

CONDITIONS OF TEST FOR A.C. CHARACTERISTICS

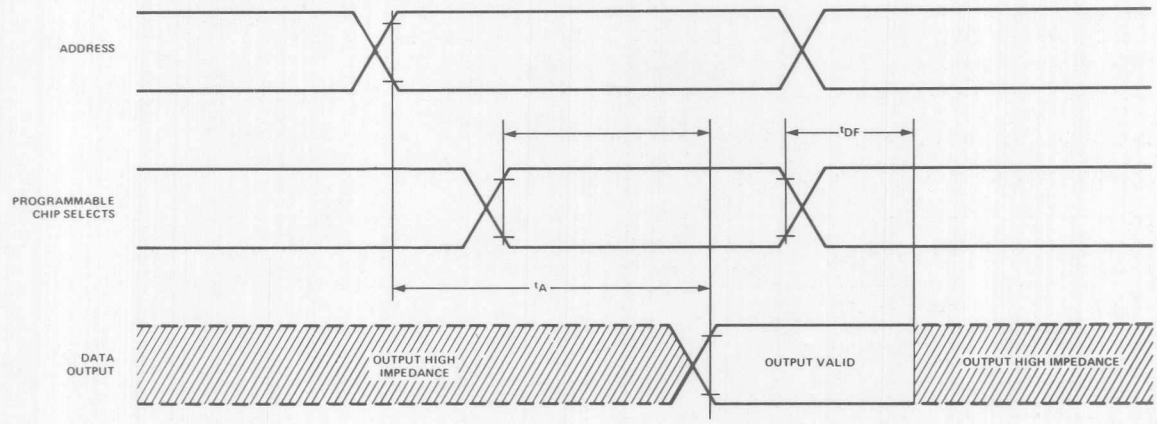
Output Load 1 TTL Gate and C_L = 100 pF
Input Pulse Levels 0.8 to 2.4V
Input Pulse Rise and Fall Times (10% to 90%) 20 ns
Timing Measurement Reference Level
Input 1V and 2.2V
Output 0.8V and 2.0V

CAPACITANCE(2) T_A = 25°C, f = 1 MHz

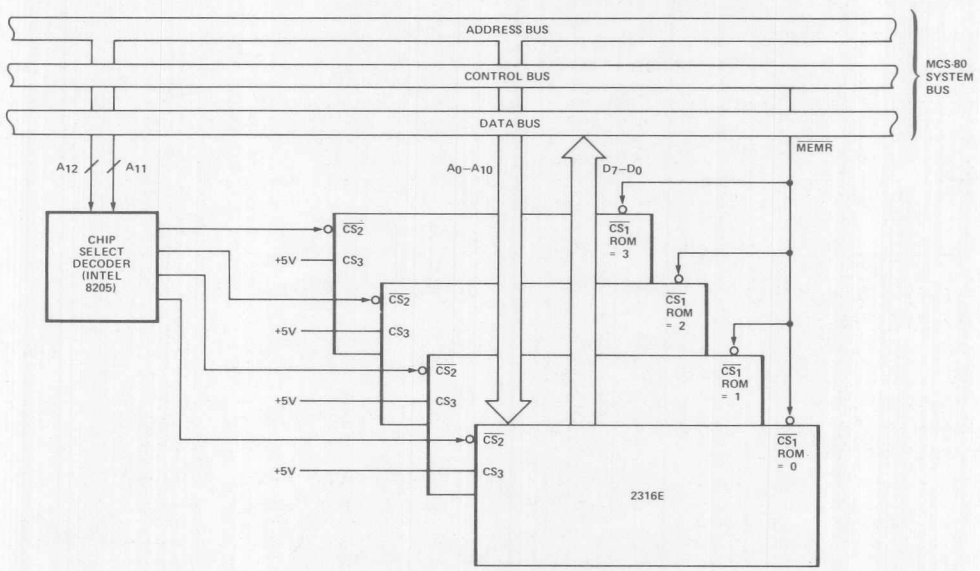
SYMBOL	TEST	LIMITS	
		TYP.	MAX.
C_{IN}	All Pins Except Pin Under Test Tied to AC Ground	5 pF	10 pF
C_{OUT}	All Pins Except Pin Under Test Tied to AC Ground	10 pF	15 pF

NOTE: 2. This parameter is periodically sampled and is not 100% tested.

A.C. Waveforms



Typical System Application (8K × 8 ROM Memory)





8708

8192 BIT ERASABLE AND ELECTRICALLY REPROGRAMMABLE READ ONLY MEMORY

• 8708 1024x8 Organization

- **Fast Programming — Typ. 100 sec. For All 8K Bits**
- **Low Power During Programming**
- **Access Time — 450 ns**
- **Standard Power Supplies — +12V, ±5V**
- **Static — No Clocks Required**
- **Inputs and Outputs TTL Compatible During Both Read and Program Modes**
- **Three-State Output — OR-Tie Capability**

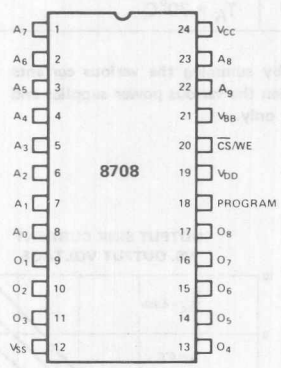
The Intel® 8708 is a high speed 8192 bit erasable and electrically reprogrammable ROM (EPROM) ideally suited where fast turn around and pattern experimentation are important requirements.

The 8708 is packaged in a 24 pin dual-in-line package with transparent lid. The transparent lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. A new pattern can then be written into the device.

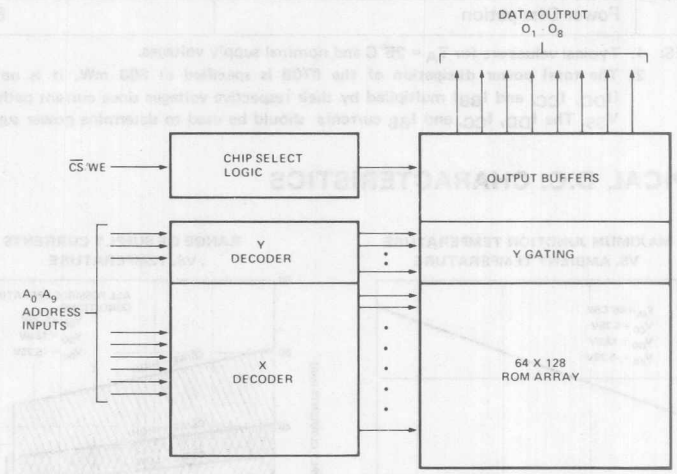
A pin for pin mask programmed ROM, the Intel® 8308, is available for large volume production runs of systems initially using the 8708.

The 8708 is fabricated with the time proven N-channel silicon gate technology.

PIN CONFIGURATION



BLOCK DIAGRAM



PIN NAMES

A_0-A_9	ADDRESS INPUTS
O_1-O_8	DATA OUTPUTS
$\overline{CS}$/WE	CHIP SELECT/WRITE ENABLE INPUT

PROGRAMMING

The programming specifications are identical to those of the 2708. (See ROM and PROM Programming Instructions, page 6-74).

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias -25°C to +85°C
Storage Temperature -65°C to +125°C
V_{DD} With Respect to V_{BB} +20V to -0.3V
V_{CC} and V_{SS} With Respect to V_{BB} +15V to -0.3V
All Input or Output Voltages With
Respect to V_{BB} During Read +15V to -0.3V
$\overline{CS}$/WE Input With Respect to V_{BB}
During Programming +20V to -0.3V
Program Input With Respect to V_{BB} +35V to -0 3V
Power Dissipation 1.5W

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

READ OPERATION

D.C. AND OPERATING CHARACTERISTICS

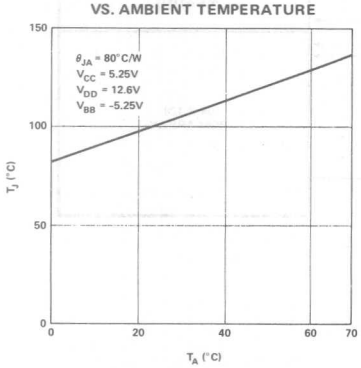
T_A = 0°C to 70°C, V_{CC} = +5V ±5%, V_{DD} = +12V ±5%, V_{BB} = -5V ±5%, V_{SS} = 0V, Unless Otherwise Noted.

Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Conditions
I_{LI}	Address and Chip Select Input Sink Current		1	10	μA	V_{IN} = 5.25 V or V_{IN} = V_{IL}
I_{LO}	Output Leakage Current		1	10	μA	V_{OUT} = 5.25V, $\overline{CS}$/WE = 5V
I_{DD}[2]	V_{DD} Supply Current		50	65	mA	Worst Case Supply Currents:
I_{CC}[2]	V_{CC} Supply Current		6	10	mA	All Inputs High
I_{BB}[2]	V_{BB} Supply Current		30	45	mA	$\overline{CS}$/WE = 5V; T_A = 0°C
V_{IL}	Input Low Voltage	V_{SS}		0.65	V	
V_{IH}	Input High Voltage	3.0		V_{CC}+1	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 1.6mA
V_{OH1}	Output High Voltage	3.7			V	I_{OH} = -100μA
V_{OH2}	Output High Voltage	2.4			V	I_{OH} = -1mA
P_D	Power Dissipation			800	mW	T_A = 70°C

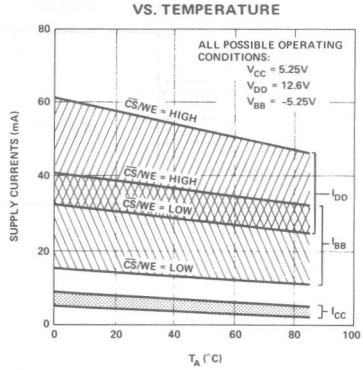
NOTES: 1. Typical values are for T_A = 25°C and nominal supply voltages.
2. The total power dissipation of the 8708 is specified at 800 mW. It is not calculable by summing the various currents (I_{DD}, I_{CC}, and I_{BB}) multiplied by their respective voltages since current paths exist between the various power supplies and V_{SS}. The I_{DD}, I_{CC}, and I_{BB} currents should be used to determine power supply capacity only.

TYPICAL D.C. CHARACTERISTICS

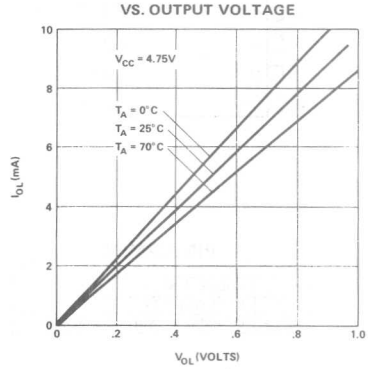
MAXIMUM JUNCTION TEMPERATURE VS. AMBIENT TEMPERATURE



RANGE OF SUPPLY CURRENTS VS. TEMPERATURE



OUTPUT SINK CURRENT VS. OUTPUT VOLTAGE



A.C. CHARACTERISTICS

$T_A = 0°C$ to $70°C$, $V_{CC} = +5V \pm 5\%$, $V_{DD} = +12V \pm 5\%$, $V_{BB} = -5V \pm 5\%$, $V_{SS} = 0V$, Unless Otherwise Noted.

Symbol	Parameter	Min.	Typ.	Max.	Unit
t_{ACC}	Address to Output Delay		280	450	ns
$t_{\overline{CO}}$	Chip Select to Output Delay		60	120	ns
t_{DF}	Chip De-Select to Output Float	0		120	ns
t_{OH}	Address to Output Hold	0			ns

CAPACITANCE $T_A = 25°C$, f = 1MHz

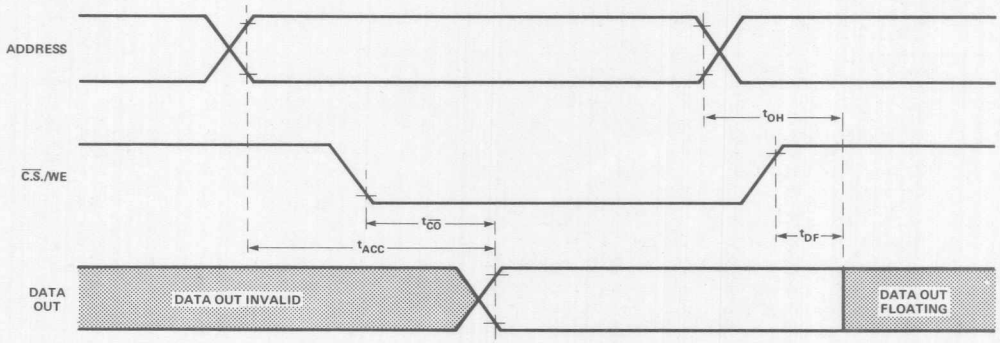
Symbol	Parameter	Typ.	Max.	Unit	Conditions
C_{IN}	Input Capacitance	4	6	pF	$V_{IN}=0V$
C_{OUT}	Output Capacitance	8	12	pF	$V_{OUT}=0V$

Note . This parameter is periodically sampled and not 100% tested.

A.C. TEST CONDITIONS

Output Load: 1 TTL gate and C_L = 100pF
Input Rise and Fall Times: ≤20ns
Timing Measurement Reference Levels: 0.8V and 2.8V for inputs; 0.8V and 2.4V for outputs
Input Pulse Levels: 0.65V to 3.0V

WAVEFORMS



ERASURE CHARACTERISTICS

The erasure characteristics of the 8708 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000-4000Å range. Data show that constant exposure to room level flourescent lighting could erase the typical 8708 in approximately 3 years while it would take approximately 1 week to cause erasure when exposed to direct sunlight. If the 8708 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8708 window to prevent unintentional erasure.

The recommended erasure procedure (see page 3-55) for the 8708 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000μW/cm^2 power rating. The 8708 should be placed within one inch from the lamp tubes during erasure. Some lamps have a filter on their tubes and this filter should be removed before erasure.

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

2716
16K (2K×8) UV ERASABLE PROM

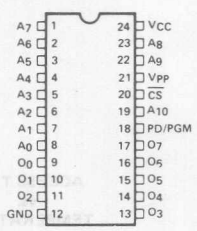
- **Single +5V Power Supply**
- **Simple Programming Requirements**
 Single Location Programming
 Programs With One 50ms Pulse
- **Low Power Dissipation**
 525mW Max. Active Power
 132mW Max. Standby Power
- **Pin Compatible To Intel 2316E ROM**
- **Fast Access Time: 450ns Max.**
- **Inputs and Outputs TTL**
 Compatible During Read
 And Program

The Intel® 2716 is a 16,384-bit ultraviolet erasable and electrically programmable read-only memory (EPROM). The 2716 operates from a single 5-volt power supply, has a static power down mode, and features fast single address location programming. It makes designing with EPROMs faster, easier and more economical. For production quantities, the 2716 user can convert rapidly to Intel's new pin-for-pin compatible 16K ROM, the 2316E.

Since the 450-nsec 2716 operates from a single 5-volt supply, it is ideal for use with the newer high performance +5V microprocessors such as Intel's 8085 and 8048. The 2716 is also the first EPROM with a static power down mode which reduces the power dissipation without increasing access time. The maximum active power dissipation is 525 mW while the maximum standby power dissipation is only 132 mW, a 75% savings.

The 2716 has the simplest and fastest method yet devised for programming EPROMs – single pulse TTL level programming. No need for high voltage pulsing because all programming controls are handled by TTL signals. Now, it is possible to program on-board, in the system, in the field. Program any location at any time – either individually, sequentially or at random, with the 2716's single address location programming. Total programming time for all 16,384 bits is only 100 seconds.

PIN CONFIGURATION



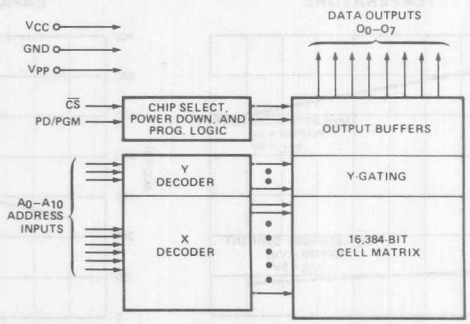
PIN NAMES

A_0–A_{10}	ADDRESSES
PD/PGM	POWER DOWN/PROGRAM
$\overline{CS}$	CHIP SELECT
O_0–O_7	OUTPUTS

MODE SELECTION

MODE \ PINS	PD/PGM (18)	$\overline{CS}$ (20)	V_{PP} (21)	V_{CC} (24)	OUTPUTS (9-11, 13-17)
Read	V_{IL}	V_{IL}	+5	+5	D_{OUT}
Deselect	Don't Care	V_{IH}	+5	+5	High Z
Power Down	V_{IH}	Don't Care	+5	+5	High Z
Program	Pulsed V_{IL} to V_{IH}	V_{IH}	+25	+5	D_{IN}
Program Verify	V_{IL}	V_{IL}	+25	+5	D_{OUT}
Program Inhibit	V_{IL}	V_{IH}	+25	+5	High Z

BLOCK DIAGRAM



PRELIMINARY
This is not a final specification. Some parameter limits are subject to change.

PROGRAMMING

The programming specifications are described in the PROM/ROM Programming Instructions on page 3-55.

Absolute Maximum Ratings*

Temperature Under Bias −10°C to +80°C
Storage Temperature −65°C to +125°C
All Input or Output Voltages with Respect to Ground +6V to −0.3V
V_{PP} Supply Voltage with Respect to Ground +28V to −0.3V

**COMMENT:* Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

READ OPERATION

D.C. and Operating Characteristics

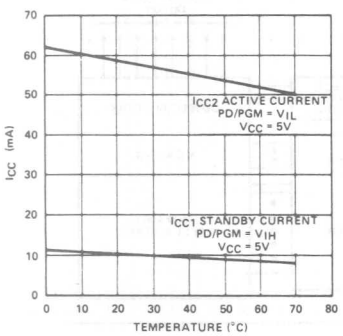
T_A = 0°C to 70°C, V_{CC}[1,2] = +5V ±5%, V_{PP}[2] = V_{CC} ±0.6V[3]

Symbol	Parameter	Limits			Unit	Conditions
		Min.	Typ.[4]	Max.		
I_{LI}	Input Load Current			10	μA	V_{IN} = 5.25V
I_{LO}	Output Leakage Current			10	μA	V_{OUT} = 5.25V
I_{PP1}[2]	V_{PP} Current			5	mA	V_{PP} = 5.85V
I_{CC1}[2]	V_{CC} Current (Standby)		10	25	mA	PD/PGM = V_{IH}, $\overline{CS}$ = V_{IL}
I_{CC2}[2]	V_{CC} Current (Active)		57	100	mA	$\overline{CS}$ = PD/PGM = V_{IL}
V_{IL}	Input Low Voltage	−0.1		0.8	V	
V_{IH}	Input High Voltage	2.2		V_{CC}+1	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 2.1 mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = −400 μA

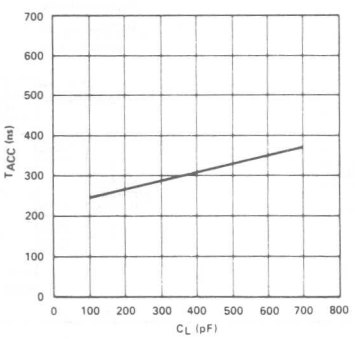
NOTES:
1. V_{CC} must be applied simultaneously or before V_{PP} and removed simultaneously or after V_{PP}.
2. V_{PP} may be connected directly to V_{CC} except during programming. The supply current would then be the sum of I_{CC} and I_{PP1}.
3. The tolerance of 0.6V allows the use of a driver circuit for switching the V_{PP} supply pin from V_{CC} in read to 25V for programming.
4. Typical values are for T_A = 25°C and nominal supply voltages.
5. This parameter is only sampled and is not 100% tested.
6. t_{ACC2} is referenced to PD/PGM or the addresses, whichever occurs last.

Typical Characteristics

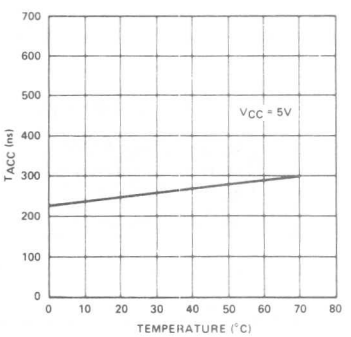
I_{CC} CURRENT vs. TEMPERATURE



ACCESS TIME vs. CAPACITANCE



ACCESS TIME vs. TEMPERATURE



PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

A.C. Characteristics

$T_A = 0°C$ to $70°C$, V_{CC}[1] = +5V ±5%, V_{PP}[2] = V_{CC} ±0.6V[3]

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.[4]	Max.		
t_{ACC1}	Address to Output Delay		250	450	ns	PD/PGM = $\overline{CS}$ = V_{IL}
t_{ACC2}	PD/PGM to Output Delay		280	450	ns	$\overline{CS}$ = V_{IL}
t_{CO}	Chip Select to Output Delay			120	ns	PD/PGM = V_{IL}
t_{PF}	PD/PGM to Output Float	0		100	ns	$\overline{CS}$ = V_{IL}
t_{DF}	Chip Deselect to Output Float	0		100	ns	PD/PGM = V_{IL}
t_{OH}	Address to Output Hold	0			ns	PD/PGM = $\overline{CS}$ = V_{IL}

Capacitance[5] $T_A = 25°C$, f = 1 MHz

Symbol	Parameter	Typ.	Max.	Unit	Conditions
C_{IN}	Input Capacitance	4	6	pF	V_{IN} = 0V
C_{OUT}	Output Capacitance	8	12	pF	V_{OUT} = 0V

NOTE: Please refer to page 2 for notes.

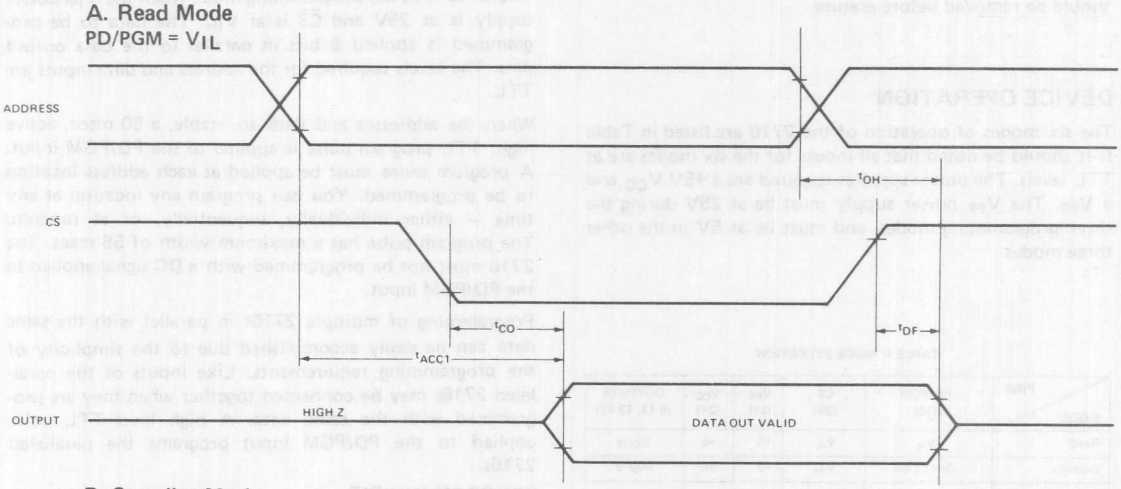
A.C. Test Conditions:

Output Load: 1 TTL gate and C_L = 100 pF
Input Rise and Fall Times: ≤20 ns
Input Pulse Levels: 0.8V to 2.2V
Timing Measurement Reference Level:
Inputs 1V and 2V
Outputs 0.8V and 2V

WAVEFORMS

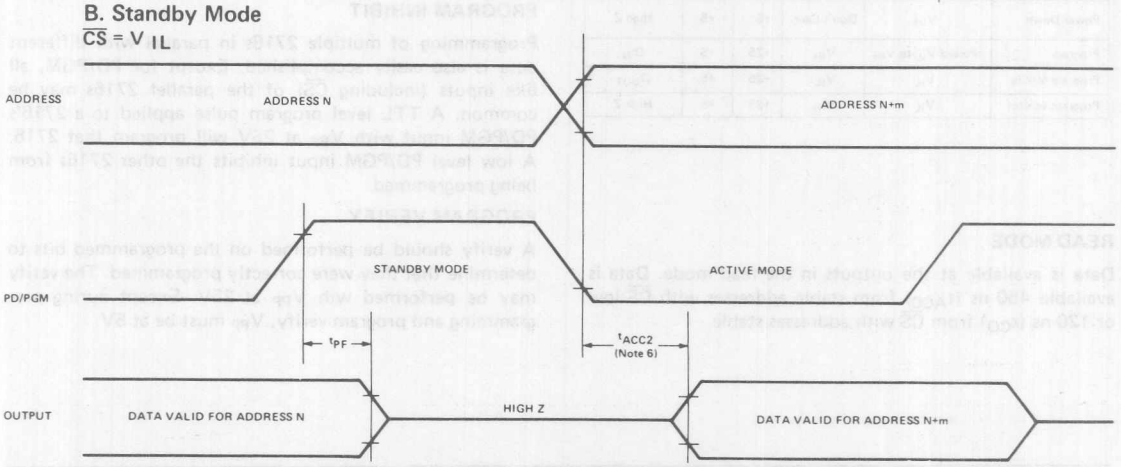
A. Read Mode

PD/PGM = V_{IL}



B. Standby Mode

$\overline{CS}$ = V_{IL}



PRELIMINARY
Notice: This is a preliminary data sheet. Parameters and specifications subject to change without notice. Contact your local Intel sales office for the latest data sheet.

ERASURE CHARACTERISTICS

The erasure characteristics of the 2716 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 2716 in approximately 3 years, while it would take approximatley 1 week to cause erasure when exposed to direct sunlight. If the 2716 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 2716 window to prevent unintentional erasure.

The recommended erasure procedure (see page 3-55) for the 2716 is exposure to shortwave ultraviolet light which has a wavelength of 2537 Angstroms (Å). The integrated dose (i.e., UV intensity X exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12000 μW/cm^2 power rating. The 2716 should be placed within 1 inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

DEVICE OPERATION

The six modes of operation of the 2716 are listed in Table I. It should be noted that all inputs for the six modes are at TTL levels. The power supplies required are a +5V V_{CC} and a V_{PP}. The V_{PP} power supply must be at 25V during the three programming modes, and must be at 5V in the other three modes.

TABLE I. MODE SELECTION

MODE \ PINS	PD/PGM (18)	$\overline{CS}$ (20)	V_{PP} (21)	V_{CC} (24)	OUTPUTS (9-11, 13-17)
Read	V_{IL}	V_{IL}	+5	+5	D_{OUT}
Deselect	Don't Care	V_{IH}	+5	+5	High Z
Power Down	V_{IH}	Don't Care	+5	+5	High Z
Program	Pulsed V_{IL} to V_{IH}	V_{IH}	+25	+5	D_{IN}
Program Verify	V_{IL}	V_{IL}	+25	+5	D_{OUT}
Program Inhibit	V_{IL}	V_{IH}	+25	+5	High Z

READ MODE

Data is available at the outputs in the read mode. Data is available 450 ns (t_{ACC}) from stable addresses with $\overline{CS}$ low or 120 ns (t_{CO}) from $\overline{CS}$ with addresses stable.

DESELECT MODE

The outputs of two or more 2716s may be OR-tied together on the same data bus. Only one 2716 should have its outputs selected ($\overline{CS}$ low) to prevent data bus contention between 2716s in this configuration. The outputs of the other 2716s should be deselected with the $\overline{CS}$ input at a high TTL level.

POWER DOWN MODE

The 2716 has a power down mode which reduces the active power dissipation by 75%, from 525 mW to 132 mW. Power down is achieved by applying a TTL high signal to the PD/PGM input. In power down the outputs are in a high impedance state, independent of the $\overline{CS}$ input.

PROGRAMMING

Initially, and after each erasure, all bits of the 2716 are in the "1" state. Data is introduced by selectively programming "0's" into the desired bit locations. Although only "0's" will be programmed, both "1's" and "0's" can be presented in the data word. The only way to change a "0" to a "1" is by ultraviolet light erasure.

The 2716 is in the programming mode when the V_{PP} power supply is at 25V and $\overline{CS}$ is at V_{IH}. The data to be programmed is applied 8 bits in parallel to the data output pins. The levels required for the address and data inputs are TTL.

When the addresses and data are stable, a 50 msec, active high, TTL program pulse is applied to the PD/PGM input. A program pulse must be applied at each address location to be programmed. You can program any location at any time — either individually, sequentially, or at random. The program pulse has a maximum width of 55 msec. The 2716 must not be programmed with a DC signal applied to the PD/PGM input.

Programming of multiple 2716s in parallel with the same data can be easily accomplished due to the simplicity of the programming requirements. Like inputs of the paralleled 2716s may be connected together when they are programmed with the same data. A high level TTL pulse applied to the PD/PGM input programs the paralleled 2716s.

PROGRAM INHIBIT

Programming of multiple 2716s in parallel with different data is also easily accomplished. Except for PD/PGM, all like inputs (including $\overline{CS}$) of the parallel 2716s may be common. A TTL level program pulse applied to a 2716's PD/PGM input with V_{PP} at 25V will program that 2716. A low level PD/PGM input inhibits the other 2716s from being programmed.

PROGRAM VERIFY

A verify should be performed on the programmed bits to determine that they were correctly programmed. The verify may be performed wth V_{PP} at 25V. Except during programming and program verify, V_{PP} must be at 5V.

8101A-4
1024 BIT STATIC MOS RAM WITH SEPARATE I/O

* 450 nsec Access Time Maximum
* 256 Word by 4 Bit Organization

- **Single +5V Supply Voltage**
- **Directly TTL Compatible: All Inputs and Outputs**
- **Static MOS: No Clocks or Refreshing Required**
- **Simple Memory Expansion: Chip Enable Input**
- **Powerful Output Drive Capability**
- **Low Cost Packaging: 22 Pin Plastic Dual In-Line Configuration**
- **Low Power: Typically 150mW**
- **Three-State Output: OR-Tie Capability**
- **Output Disable Provided for Ease of Use in Common Data Bus Systems**

The Intel® 8101A-4 is a 256 word by 4-bit static random access memory element using N-channel MOS devices integrated on a monolithic array. It uses fully DC stable (static) circuitry and therefore requires no clocks or refreshing to operate. The data is read out nondestructively and has the same polarity as the input data.

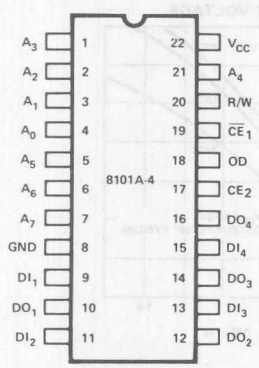
The 8101A-4 is designed for memory applications where high performance, low cost, large bit storage, and simple interfacing are important design objectives.

It is directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. Two chip-enables allow easy selection of an individual package when outputs are OR-tied. An output disable is provided so that data inputs and outputs can be tied for common I/O systems. The output disable function eliminates the need for bi-directional logic in a common I/O system.

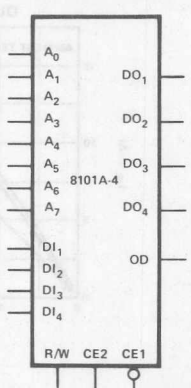
The Intel® 8101A-4 is fabricated with N-channel silicon gate technology. This technology allows the design and production of high performance, easy-to-use MOS circuits and provides a higher functional density on a monolithic chip than either conventional MOS technology or P-channel silicon gate technology.

Intel's silicon gate technology also provides excellent protection against contamination. This permits the use of low cost plastic packaging.

PIN CONFIGURATION



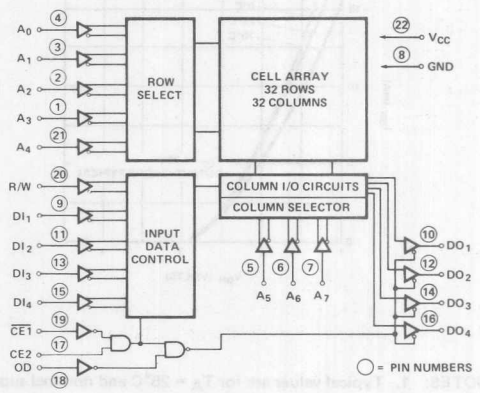
LOGIC SYMBOL



PIN NAMES

DI_1-DI_4	DATA INPUT	$\overline{CE}_2$	CHIP ENABLE 2
A_0-A_7	ADDRESS INPUTS	OD	OUTPUT DISABLE
R/W	READ/WRITE INPUT	DO_1-DO_4	DATA OUTPUT
$\overline{CE}_1$	CHIP ENABLE 1	V_{CC}	POWER (+5V)

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias -10°C to 80°C

Storage Temperature -65°C to +150°C

Voltage On Any Pin
With Respect to Ground -0.5V to +7V

Power Dissipation 1 Watt

*COMMENT:

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

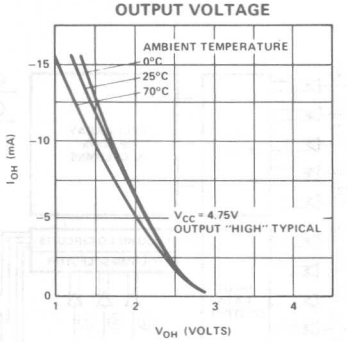
D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = 5V ±5% unless otherwise specified.

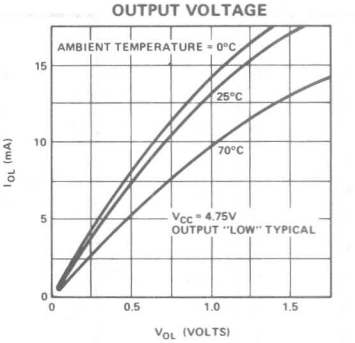
Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Test Conditions
I_{LI}	Input Current		1	10	μA	V_{IN} = 0 to 5.25V
I_{LOH}	I/O Leakage Current[2]		1	10	μA	Output Disabled, V_{OUT}=4.0V
I_{LOL}	I/O Leakage Current[2]		-1	-10	μA	Output Disabled, V_{OUT}=0.45V
I_{CC1}	Power Supply Current		35	55	mA	V_{IN} = 5.25V, I_O = 0mA T_A = 25°C
I_{CC2}	Power Supply Current			60	mA	V_{IN} = 5.25V, I_O = 0mA T_A = 0°C
V_{IL}	Input "Low" Voltage	-0.5		+0.8	V	
V_{IH}	Input "High" Voltage	2.0		V_{CC}	V	
V_{OL}	Output "Low" Voltage			+0.45	V	I_{OL} = 2.0mA
V_{OH}	Output "High" Voltage	2.4			V	I_{OH} = -400μA

TYPICAL D.C. CHARACTERISTICS

OUTPUT SOURCE CURRENT VS. OUTPUT VOLTAGE



OUTPUT SINK CURRENT VS. OUTPUT VOLTAGE



NOTES: 1. Typical values are for T_A = 25°C and nominal supply voltage.
2. Input and Output tied together.

A.C. CHARACTERISTICS

READ CYCLE $T_A = 0°C$ to $70°C$, $V_{CC} = 5V \pm 5\%$, unless otherwise specified.

Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Test Conditions
t_{RC}	Read Cycle	450			ns	(See Below)
t_A	Access Time			450	ns	
t_{CO}	Chip Enable To Output			310	ns	
t_{OD}	Output Disable To Output			250	ns	
t_{DF} [2]	Data Output to High Z State	0		200	ns	
t_{OH}	Previous Read Data Valid after change of Address	40			ns	

WRITE CYCLE

Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Test Conditions
t_{WC}	Write Cycle	270			ns	(See Below)
t_{AW}	Write Delay	20			ns	
t_{CW}	Chip Enable To Write	250			ns	
t_{DW}	Data Setup	250			ns	
t_{DH}	Data Hold	0			ns	
t_{WP}	Write Pulse	250			ns	
t_{WR}	Write Recovery	0			ns	
t_{DS}	Output Disable Setup	20			ns	

A.C. CONDITIONS OF TEST

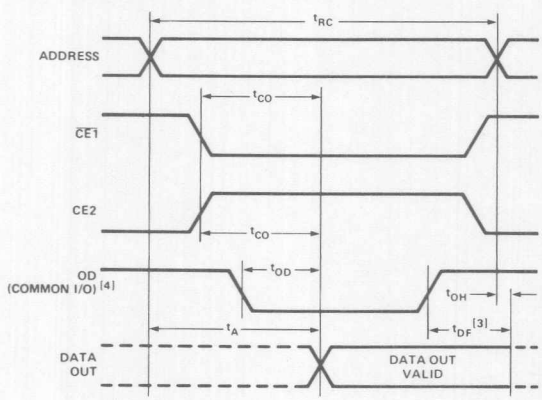
t_r, t_f 20 ns
Input Levels 0.8V or 2.0V
Timing Reference 1.5V
Load 1 TTL Gate and C_L = 100 pF

CAPACITANCE[3] $T_A = 25°C$, f = 1 MHz

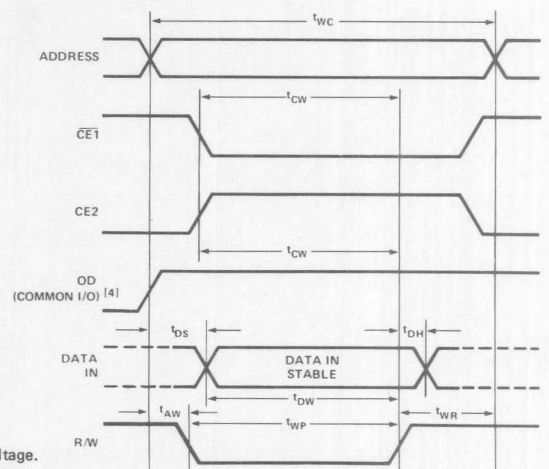
Symbol	Test	Limits (pF)	
		Typ.[1]	Max.
C_{IN}	Input Capacitance (All Input Pins) $V_{IN} = 0V$	4	8
C_{OUT}	Output Capacitance $V_{OUT} = 0V$	8	12

WAVEFORMS

READ CYCLE



WRITE CYCLE



NOTES:
1. Typical values are for $T_A = 25°C$ and nominal supply voltage.
2. t_{DF} is with respect to the trailing edge of $\overline{CE}_1$, CE_2, or OD, whichever occurs first.
3. This parameter is periodically sampled and is not 100% tested.
4. OD should be tied low for separate I/O operation.

8111A-4
1024 BIT STATIC MOS RAM WITH COMMON I/O

* 450 nsec Access Time Maximum
* 256 Word by 4 Bit Organization

- **Single +5V Supply Voltage**
- **Directly TTL Compatible: All Inputs and Outputs**
- **Static MOS: No Clocks or Refreshing Required**
- **Simple Memory Expansion: Chip Enable Input**
- **Powerful Output Drive Capability**
- **Low Cost Packaging: 18 Pin Plastic Dual In-Line Configuration**
- **Low Power: Typically 150mW**
- **Three-State Output: OR-Tie Capability**
- **Output Disable Provided for Ease of Use in Common Data Bus Systems**

The Intel® 8111A-4 is a 256 word by 4-bit static random access memory element using N-channel MOS devices integrated on a monolithic array. It uses fully DC stable (static) circuitry and therefore requires no clocks or refreshing to operate. The data is read out nondestructively and has the same polarity as the input data. Common input/output pins are provided.

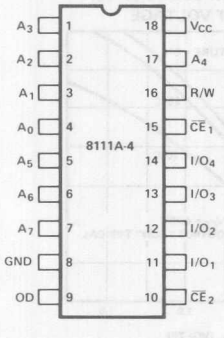
The 8111A-4 is designed for memory applications in small systems where high performance, low cost, large bit storage, and simple interfacing are important design objectives.

It is directly TTL compatible in all respects: inputs, outputs, and a single +5V supply. Separate chip enable ($\overline{CE}$) leads allow easy selection of an individual package when outputs are OR-tied.

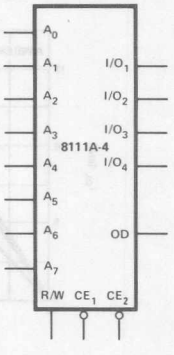
The Intel® 8111A-4 is fabricated with N-channel silicon gate technology. This technology allows the design and production of high performance, easy-to-use MOS circuits and provides a higher functional density on a monolithic chip than either conventional MOS technology or P-channel silicon gate technology.

Intel's silicon gate technology also provides excellent protection against contamination. This permits the use of low cost plastic packaging.

PIN CONFIGURATION



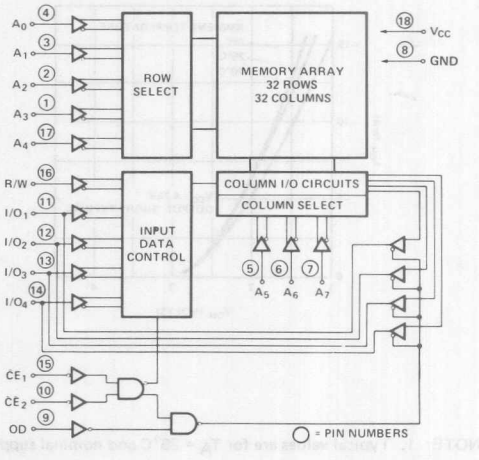
LOGIC SYMBOL



PIN NAMES

A_0-A_7	ADDRESS INPUTS
OD	OUTPUT DISABLE
R/W	READ/WRITE INPUT
$\overline{CE}_1$	CHIP ENABLE 1
$\overline{CE}_2$	CHIP ENABLE 2
I/O_1- I/O_4	DATA INPUT/OUTPUT

BLOCK DIAGRAM



ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias -10°C to 80°C

Storage Temperature -65°C to +150°C

Voltage On Any Pin
With Respect to Ground -0.5V to +7V

Power Dissipation 1 Watt

*COMMENT:

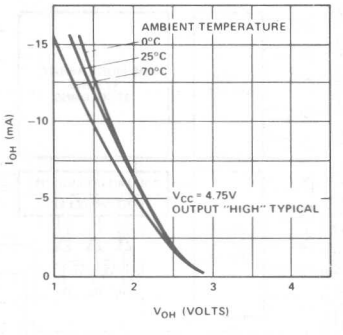
Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS

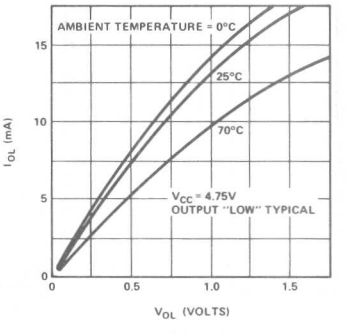
T_A = 0°C to 70°C, V_{CC} = 5V ±5% , unless otherwise specified.

Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Test Conditions
I_{LI}	Input Load Current		1	10	µA	V_{IN} = 0 to 5.25V
I_{LOH}	I/O Leakage Current		1	10	µA	Output Disabled, $V_{I/O}$=4.0V
I_{LOL}	I/O Leakage Current		-1	-10	µA	Output Disabled, $V_{I/O}$=0.45V
I_{CC1}	Power Supply Current		35	55	mA	V_{IN} = 5.25V $I_{I/O}$ = 0mA, T_A = 25°C
I_{CC2}	Power Supply Current			60	mA	V_{IN} = 5.25V $I_{I/O}$ = 0mA, T_A = 0°C
V_{IL}	Input Low Voltage	-0.5		0.8	V	
V_{IH}	Input High Voltage	2.0		V_{CC}	V	
V_{OL}	Output Low Voltage			0.45	V	I_{OL} = 2.0mA
V_{OH}	Output High Voltage	2.4			V	I_{OH} = -400µA

OUTPUT SOURCE CURRENT VS. OUTPUT VOLTAGE



OUTPUT SINK CURRENT VS. OUTPUT VOLTAGE



NOTE: 1. Typical values are for T_A = 25°C and nominal supply voltage.

A.C. CHARACTERISTICS

READ CYCLE $T_A = 0°C$ to $70°C$, $V_{CC} = 5V \pm 5\%$, unless otherwise specified.

Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Test Conditions
t_{RC}	Read Cycle	450			ns	(See Below)
t_A	Access Time			450	ns	
t_{CO}	Chip Enable To Output			310	ns	
t_{OD}	Output Disable To Output			250	ns	
t_{DF} [2]	Data Output to High Z State	0		200	ns	
t_{OH}	Previous Read Data Valid after change of Address	40			ns	

WRITE CYCLE

Symbol	Parameter	Min.	Typ.[1]	Max.	Unit	Test Conditions
t_{WC}	Write Cycle	270			ns	(See Below)
t_{AW}	Write Delay	20			ns	
t_{CW}	Chip Enable To Write	250			ns	
t_{DW}	Data Setup	250			ns	
t_{DH}	Data Hold	0			ns	
t_{WP}	Write Pulse	250			ns	
t_{WR}	Write Recovery	0			ns	
t_{DS}	Output Disable Setup	20			ns	

A.C. CONDITIONS OF TEST

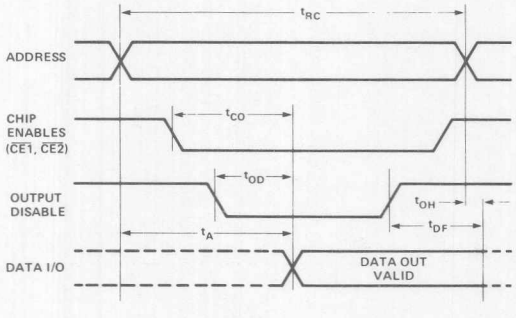
t_r, t_f 20 ns
Input Levels 0.8V or 2.0V
Timing Reference 1.5V
Load 1 TTL Gate and C_L = 100 pF

CAPACITANCE[3] $T_A = 25°C$, f = 1MHz

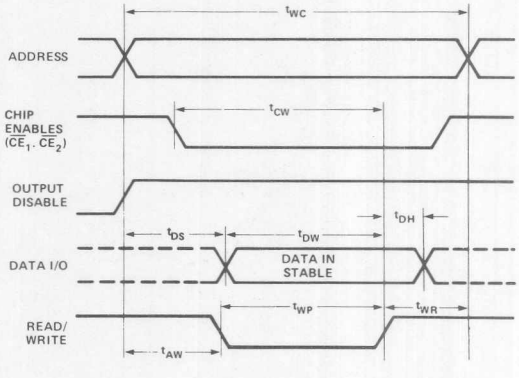
Symbol	Test	Limits (pF)	
		Typ.[1]	Max.
C_{IN}	Input Capacitance (All Input Pins) V_{IN} = 0V	4	8
$C_{I/O}$	I/O Capacitance $V_{I/O}$ = 0V	10	15

WAVEFORMS

READ CYCLE



WRITE CYCLE



NOTES: 1. Typical values are for $T_A = 25°C$ and nominal supply voltage.
2. t_{DF} is with respect to the trailing edge of $\overline{CE_1}$, $\overline{CE_2}$, or OD, whichever occurs first.
3. This parameter is periodically sampled and is not 100% tested.

5101 FAMILY
256 X 4 BIT STATIC CMOS RAM

P/N	Typ. Current @ 2V (μA)	Typ. Current @ 5V (μA)	Max Access (ns)
5101L	0.14	0.2	650
5101L-1	0.14	0.2	450
5101L-3	0.70	1.0	650
5101-8	—	10.0	800

- **Single +5V Power Supply**
- **Ideal for Battery Operation (5101L)**
- **Directly TTL Compatible: All Inputs and Outputs**
- **Three-State Output**

The Intel® 5101 is an ultra-low power 1024-bit (256 words X 4 bits) static RAM fabricated with an advanced ion-implanted silicon gate CMOS technology. The device has two chip enable inputs. Minimum standby current is drawn by this device when CE2 is at a low level. When deselected the 5101 draws from the single 5-volt supply only 10 microamps. This device is ideally suited for low power applications where battery operation or battery backup for non-volatility are required.

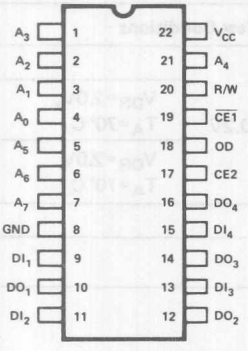
The 5101 uses fully DC stable (static) circuitry; it is not necessary to pulse chip select for each address transition. The data is read out non-destructively and has the same polarity as the input data. All inputs and outputs are directly TTL compatible. The 5101 has separate data input and data output terminals. An output disable function is provided so that the data inputs and outputs may be wire OR-ed for use in common data I/O systems.

The 5101L has the additional feature of guaranteed data retention at a power supply voltage as low as 2.0 volts.

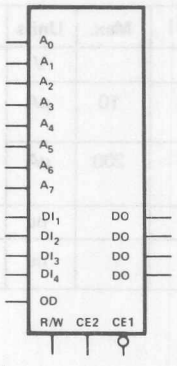
A pin compatible N-channel static RAM, the Intel® 2101A, is also available for low cost applications where a 256 X 4 organization is needed.

The Intel ion-implanted, silicon gate, Complementary MOS (CMOS) process allows the design and production of ultra-low power, high performance memories.

PIN CONFIGURATION



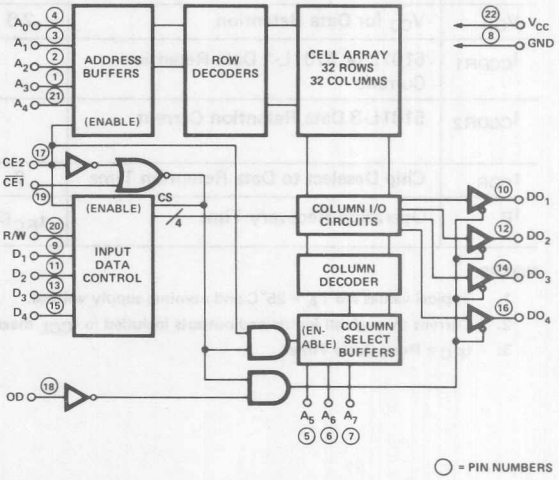
LOGIC SYMBOL



TRUTH TABLE

$\overline{CE_1}$	CE_2	OD	R/W	D_{IN}	Output	Mode
H	X	X	X	X	High Z	Not Selected
X	L	X	X	X	High Z	Not Selected
X	X	H	H	X	High Z	Output Disabled
L	H	H	L	X	High Z	Write
L	H	L	L	X	D_{IN}	Write
L	H	L	H	X	D_{OUT}	Read

BLOCK DIAGRAM



Absolute Maximum Ratings *

Ambient Temperature Under Bias–10°C to 80°C
Storage Temperature -65°C to +150°C
Voltage On Any Pin
With Respect to Ground -0.3V to V_{CC} +0.3V
Maximum Power Supply Voltage +7.0V
Power Dissipation 1 Watt

*COMMENT:

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D. C. and Operating Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ±5% unless otherwise specified.

Symbol	Parameter	5101L and 5101L-1 Limits Min.	Typ.[1]	Max.	5101L-3 Limits Min.	Typ.[1]	Max.	5101-8 Limits Min.	Typ.[1]	Max.	Units	Test Conditions
I_{L2}[2]	Input Current		5			5			5		nA	
$\|I_{LO}\|$[2]	Output Leakage Current			1			1			2	μA	$\overline{CE1}$=2.2V, V_{OUT}= 0 to V_{CC}
I_{CC1}	Operating Current		9	22		9	22		11	25	mA	V_{IN}=V_{CC}, Except $\overline{CE1} \leqslant$ 0.65V, Outputs Open
I_{CC2}	Operating Current		13	27		13	27		15	30	mA	V_{IN}=2.2V, Except $\overline{CE1} \leqslant$ 0.65V, Outputs Open
I_{CCL}[2]	Standby Current			10			200			500	μA	CE2 ⩽ 0.2V, T_A= 70° C
V_{IL}	Input Low Voltage	-0.3		0.65	-0.3		0.65	-0.3		0.65	V	
V_{IH}	Input High Voltage	2.2		V_{CC}	2.2		V_{CC}	2.2		V_{CC}	V	
V_{OL}	Output Low Voltage			0.4			0.4			0.4	V	I_{OL}=2.0 mA
V_{OH}	Output High Voltage	2.4			2.4			2.4			V	I_{OH}= -1.0 mA

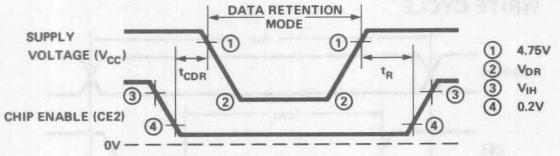
Low V_{CC} Data Retention Characteristics (For 5101L, 5101L-1 and 5101L-3) T_A = 0° C to 70° C

Symbol	Parameter	Min.	Typ.[1]	Max.	Units	Test Conditions	
V_{DR}	V_{CC} for Data Retention	2.0			V	CE2 ⩽ 0.2V	
I_{CCDR1}	5101L or 5101L-1 Data Retention Current		0.14	10	μA		V_{DR}=2.0V, T_A=70° C
I_{CCDR2}	5101L-3 Data Retention Current		0.70	200	μA		V_{DR}=2.0V, T_A=70° C
t_{CDR}	Chip Deselect to Data Retention Time	0			ns		
t_R	Operation Recovery Time	t_{RC}[3]			ns		

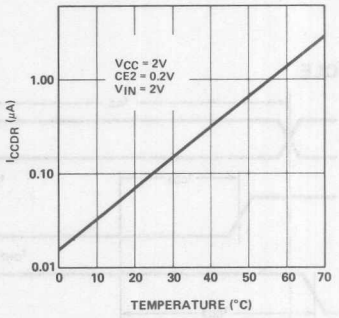
NOTES:

1. Typical values are T_A = 25°C and nominal supply voltage.
2. Current through all inputs and outputs included in I_{CCL} measurement.
3. t_{RC} = Read Cycle Time.

Low V_{CC} Data Retention Waveform



Typical I_{CCDR} Vs. Temperature



A.C. Characteristics

T_A = 0°C to 70°C, V_{CC} = 5V ±5%, unless otherwise specified.

READ CYCLE

Symbol	Parameter	5101L-1 Limits (ns) Min.	5101L-1 Limits (ns) Max.	5101L and 5101L-3 Limits (ns) Min.	5101L and 5101L-3 Limits (ns) Max.	5101-8 Limits (ns) Min.	5101-8 Limits (ns) Max.
t_{RC}	Read Cycle	450		650		800	
t_A	Access Time		450		650		800
t_{CO1}	Chip Enable ($\overline{CE\ 1}$) to Output		400		600		800
t_{CO2}	Chip Enable (CE 2) to Output		500		700		850
t_{OD}	Output Disable to Output		250		350		450
t_{DF}	Data Output to High Z State	0	130	0	150	0	200
t_{OH1}	Previous Read Data Valid with Respect to Address Change	0		0		0	
t_{OH2}	Previous Read Data Valid with Respect to Chip Enable	0		0		0	

WRITE CYCLE

t_{WC}	Write Cycle	450		650		800	
t_{AW}	Write Delay	130		150		200	
t_{CW1}	Chip Enable ($\overline{CE\ 1}$) to Write	350		550		650	
t_{CW2}	Chip Enable (CE 2) to Write	350		550		650	
t_{DW}	Data Setup	250		400		450	
t_{DH}	Data Hold	50		100		100	
t_{WP}	Write Pulse	250		400		450	
t_{WR}	Write Recovery	50		50		100	
t_{DS}	Output Disable Setup	130		150		200	

A. C. CONDITIONS OF TEST

Input Pulse Levels: +0.65 Volt to 2.2 Volt

Input Pulse Rise and Fall Times: 20nsec

Timing Measurement Reference Level: 1.5 Volt

Output Load: 1 TTL Gate and C_L = 100pF

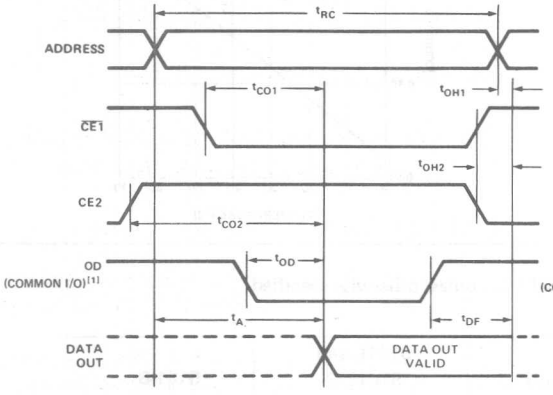
Capacitance[2] T_A = 25°C, f = 1MHz

Symbol	Test	Limits (pF) Typ.	Limits (pF) Max.
C_{IN}	Input Capacitance (All Input Pins) V_{IN} = 0V	4	8
C_{OUT}	Output Capacitance V_{OUT} = 0V	8	12

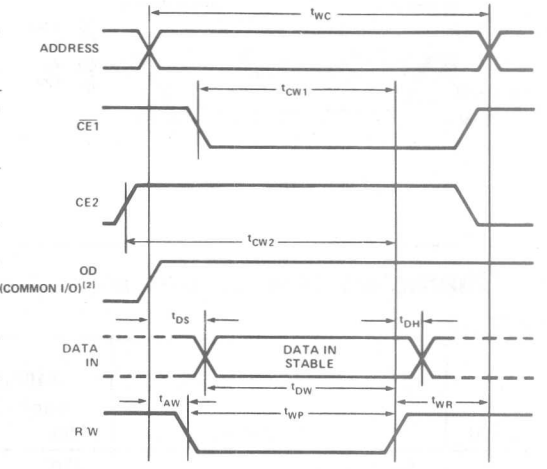
NOTES: 1. Typical values are for T_A = 25°C and nominal supply voltage.
2. This parameter is periodically sampled and is not 100% tested.

Waveforms

READ CYCLE



WRITE CYCLE



NOTES:

1. OD may be tied low for separate I/O operation.
2. During the write cycle, OD is "high" for common I/O and "don't care" for separate I/O operation.

8212

8-BIT INPUT/OUTPUT PORT

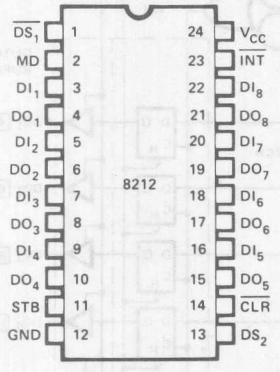
- **Fully Parallel 8-Bit Data Register and Buffer**
- **Service Request Flip-Flop for Interrupt Generation**
- **Low Input Load Current — 0.25 mA Max**
- **3-State Outputs**
- **Outputs Sink 15 mA**
- **3.65V Output High Voltage for Direct Interface to 8080 CPU or 8008 CPU**
- **Asynchronous Register Clear**
- **Replaces Buffers, Latches, and Multiplexers in Microcomputer Systems**
- **Reduces System Package Count**

The Intel® 8212 input/output port consists of an 8-bit latch with 3-state output buffers along with control and device selection logic. Also included is a service request flip-flop for the generation and control of interrupts to the microprocessor.

The device is multimode in nature. It can be used to implement latches, gated buffers or multiplexers. Thus, all of the principal peripheral and input/output functions of a microcomputer system can be implemented with this device.

*Note: The specifications for the 3212 are identical with those for the 8212.

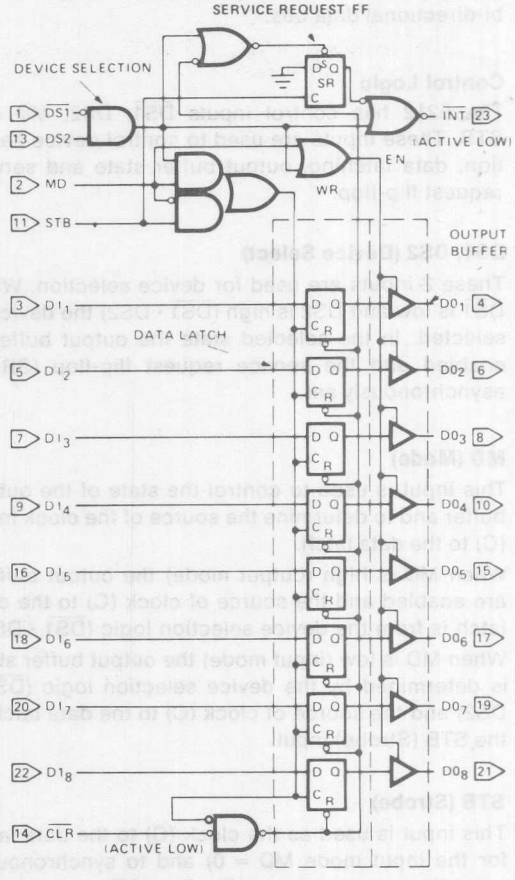
PIN CONFIGURATION



PIN NAMES

DI_1-DI_8	DATA IN
DO_1-DO_8	DATA OUT
$\overline{DS_1}$-DS_2	DEVICE SELECT
MD	MODE
STB	STROBE
$\overline{INT}$	INTERRUPT (ACTIVE LOW)
$\overline{CLR}$	CLEAR (ACTIVE LOW)

LOGIC DIAGRAM



© Intel Corporation, 1978

FUNCTIONAL DESCRIPTION

Data Latch

The 8 flip-flops that make up the data latch are of a "D" type design. The output (Q) of the flip-flop will follow the data input (D) while the clock input (C) is high. Latching will occur when the clock (C) returns low.

The data latch is cleared by an asynchronous reset input ($\overline{CLR}$). (Note: Clock (C) Overides Reset ($\overline{CLR}$).)

Output Buffer

The outputs of the data latch (Q) are connected to 3-state, non-inverting output buffers. These buffers have a common control line (EN); this control line either enables the buffer to transmit the data from the outputs of the data latch (Q) or disables the buffer, forcing the output into a high impedance state. (3-state)

This high-impedance state allows the designer to connect the 8212 directly onto the microprocessor bi-directional data bus.

Control Logic

The 8212 has control inputs $\overline{DS1}$, DS2, MD and STB. These inputs are used to control device selection, data latching, output buffer state and service request flip-flop.

DS1, DS2 (Device Select)

These 2 inputs are used for device selection. When $\overline{DS1}$ is low and DS2 is high ($\overline{DS1} \cdot DS2$) the device is selected. In the selected state the output buffer is enabled and the service request flip-flop (SR) is asynchronously set.

MD (Mode)

This input is used to control the state of the output buffer and to determine the source of the clock input (C) to the data latch.

When MD is high (output mode) the output buffers are enabled and the source of clock (C) to the data latch is from the device selection logic ($\overline{DS1} \cdot DS2$).

When MD is low (input mode) the output buffer state is determined by the device selection logic ($\overline{DS1} \cdot DS2$) and the source of clock (C) to the data latch is the STB (Strobe) input.

STB (Strobe)

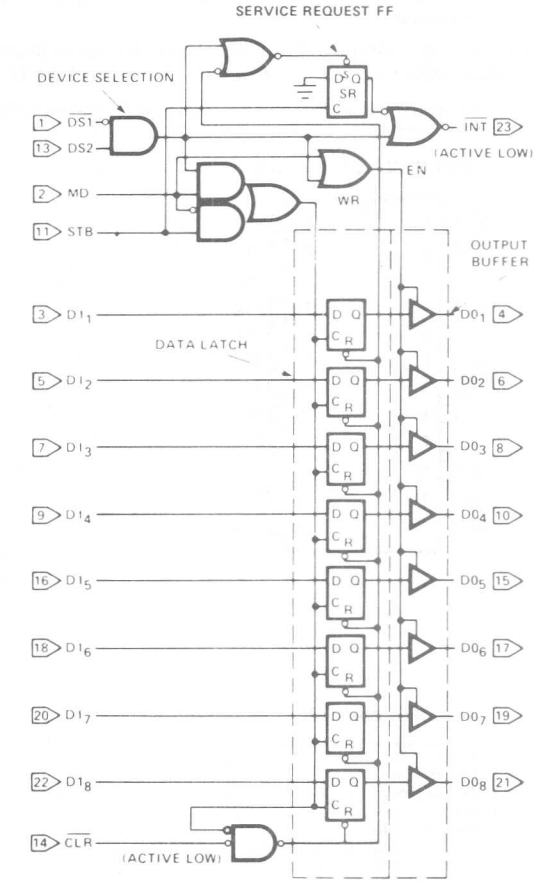
This input is used as the clock (C) to the data latch for the input mode MD = 0) and to synchronously reset the service request flip-flop (SR).

Note that the SR flip-flop is negative edge triggered.

Service Reqeust Flip-Flop

The (SR) flip-flop is used to generate and control interrupts in microcomputer systems. It is asynchronously set by the $\overline{CLR}$ input (active low). When the (SR) flip-flop is set it is in the non-interrupting state.

The output of the (SR) flip-flop (Q) is connected to an inverting input of a "NOR" gate. The other input to the "NOR" gate is non-inverting and is connected to the device selection logic ($\overline{DS1} \cdot DS2$). The output of the "NOR" gate ($\overline{INT}$) is active low (interrupting state) for connection to active low input priority generating circuits.



STB	MD	$(DS_1 \cdot DS_2)$	DATA OUT EQUALS
0	0	0	3 STATE
1	0	0	3 STATE
0	1	0	DATA LATCH
1	1	0	DATA LATCH
0	0	1	DATA LATCH
1	0	1	DATA IN
0	1	1	DATA IN
1	1	1	DATA IN

CLR	$(\overline{DS_1} \cdot DS_2)$	STB	*SR	INT
0	0	0	1	1
0	1	0	1	0
1	1	↓	0	0
1	1	0	1	0
1	0	0	1	0
1	1	↓	1	0

*INTERNAL SR FLIP-FLOP

$\overline{CLR}$ – RESETS DATA LATCH
SETS SR FLIP-FLOP
(NO EFFECT ON OUTPUT BUFFER)

Figure 1. Service Flip-Flop Function

APPLICATIONS OF THE 8212 — FOR MICROCOMPUTER SYSTEMS

- Basic schematic symbols
- Gated buffer
- Bidirectional bus driver
- Interrupting input port
- Interrupt instruction port
- Output port
- 8080A status latch
- 8085A address latch

Basic Schematic Symbols

Two examples of ways to draw the 8212 on system schematics—(1) the top being the detailed view showing pin numbers, and (2) the bottom being the symbolic view showing the system input or output as a system bus (bus containing 8 parallel lines). The output to the data bus is symbolic in referencing 8 parallel lines.

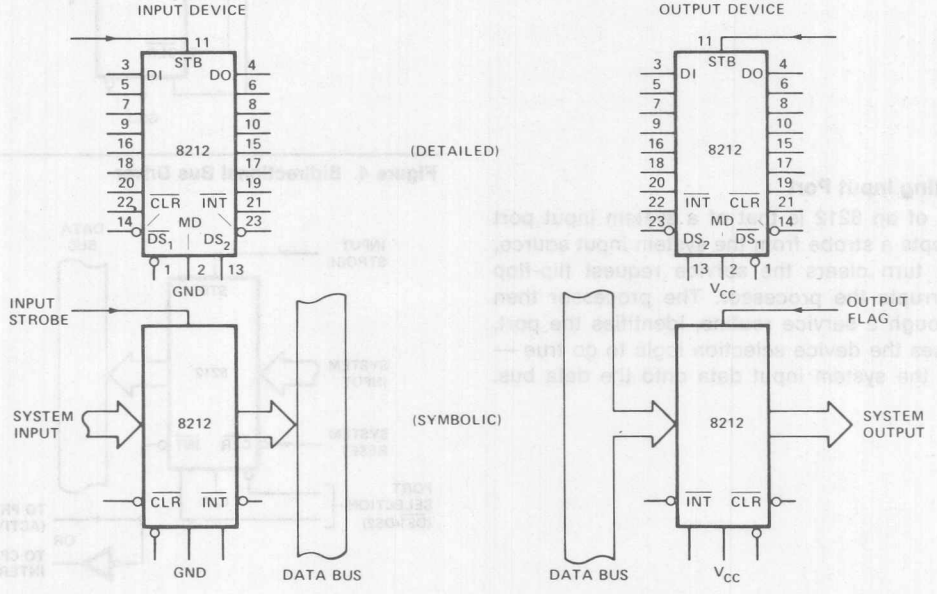


Figure 2. Basic Schematic Symbols

Gated Buffer (3-State)

The simplest use of the 8212 is that of a gated buffer. By tying the mode signal low and the strobe input high, the data latch is acting as a straight through gate. The output buffers are then enabled from the device selection logic $\overline{DS1}$ and DS2.

When the device selection logic is false, the outputs are 3-state.

When the device selection logic is true, the input data from the system is directly transferred to the output. The input data load is 250 micro amps. The output data can sink 15 milli amps. The minimum high output is 3.65 volts.

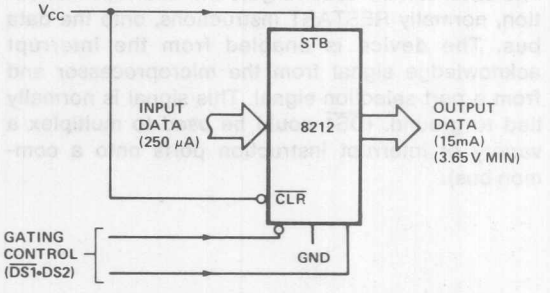


Figure 3. Gated Buffer (3-State)

Bidirectional Bus Driver

A pair of 8212's wired (back-to-back) can be used as a symmetrical drive, bi-directional bus driver. The devices are controlled by the data bus input control which is connected to $\overline{DS1}$ on the first 8212 and to DS2 on the second. One device is active, and acting as a straight through buffer the other is in 3-state mode. This is a very useful circuit in small system design.

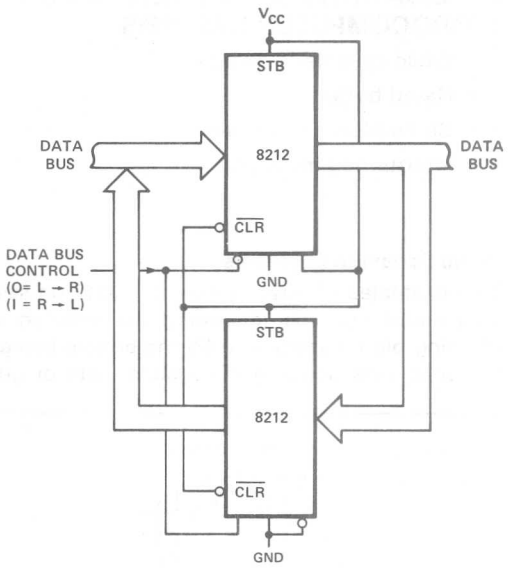


Figure 4. Bidirectional Bus Driver

Interrupting Input Port

This use of an 8212 is that of a system input port that accepts a strobe from the system input source, which in turn clears the service request flip-flop and interrupts the processor. The processor then goes through a service routine, identifies the port, and causes the device selection logic to go true — enabling the system input data onto the data bus.

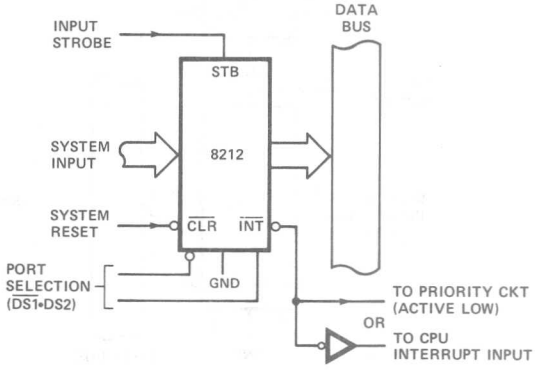


Figure 5. Interrupting Input Port

Interrupt Instruction Port

The 8212 can be used to gate the interrupt instruction, normally RESTART instructions, onto the data bus. The device is enabled from the interrupt acknowledge signal from the microprocessor and from a port selection signal. This signal is normally tied to ground. ($\overline{DS1}$ could be used to multiplex a variety of interrupt instruction ports onto a common bus).

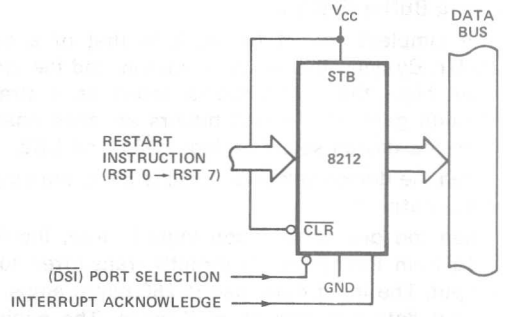


Figure 6. Interrupt Instruction Port

Output Port (With Handshaking)

The 8212 can be used to transmit data from the data bus to a system output. The output strobe could be a hand-shaking signal such as "reception of data" from the device that the system is outputting to. It in turn, can interrupt the system signifying the reception of data. The selection of the port comes from the device selection logic. ($\overline{DS1} \cdot DS2$)

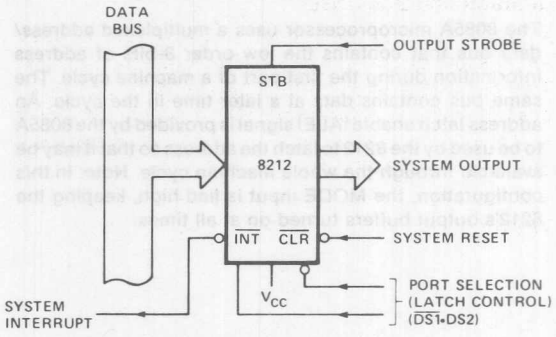


Figure 7. Output Port (With Handshaking)

8080 Status Latch

Here the 8212 is used as the status latch for an 8080 microcomputer system. The input to the 8212 latch is directly from the 8080 data bus. Timing shows that when the SYNC signal is true, which is connected to the DS2 input and the phase 1 signal is true, which is a TTL level coming from the clock generator; then, the status data will be latched into the 8212.

Note: The mode signal is tied high so that the output on the latch is active and enabled all the time.

It is shown that the two areas of concern are the bidirectional data bus of the microprocessor and the control bus.

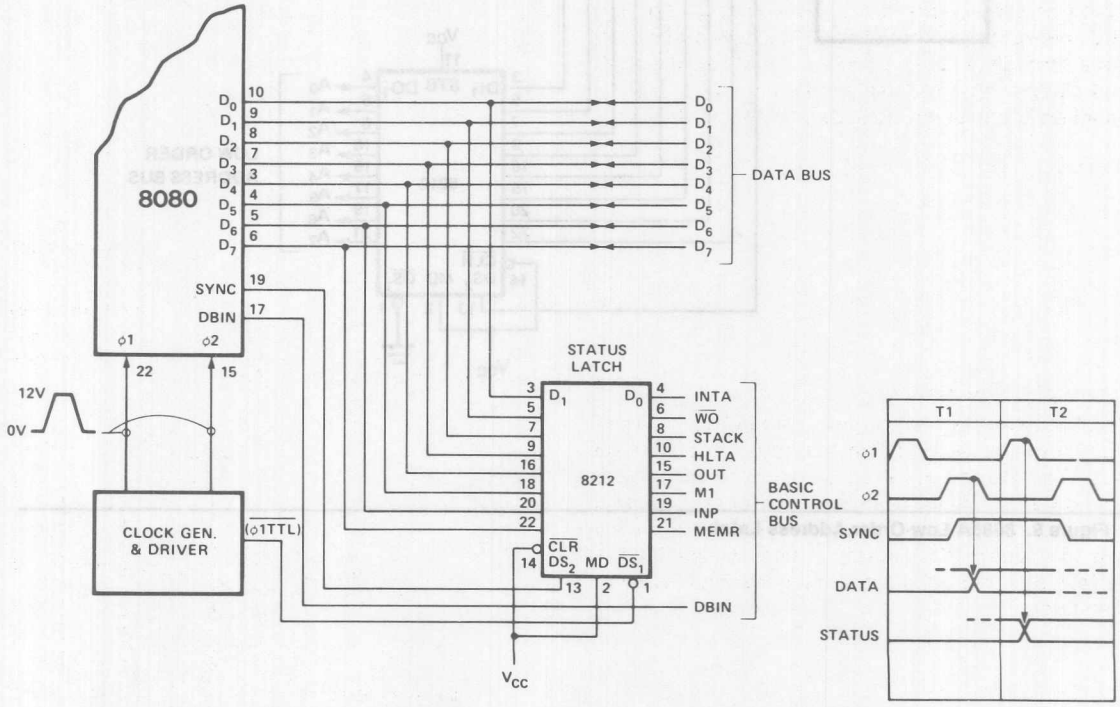


Figure 8. 8080 Status Latch

8085A Low-Order Address Latch

The 8085A microprocessor uses a multiplexed address/data bus that contains the low order 8-bits of address information during the first part of a machine cycle. The same bus contains data at a later time in the cycle. An address latch enable (ALE) signal is provided by the 8085A to be used by the 8212 to latch the address so that it may be available through the whole machine cycle. Note: In this configuration, the MODE input is tied high, keeping the 8212's output buffers turned on at all times.

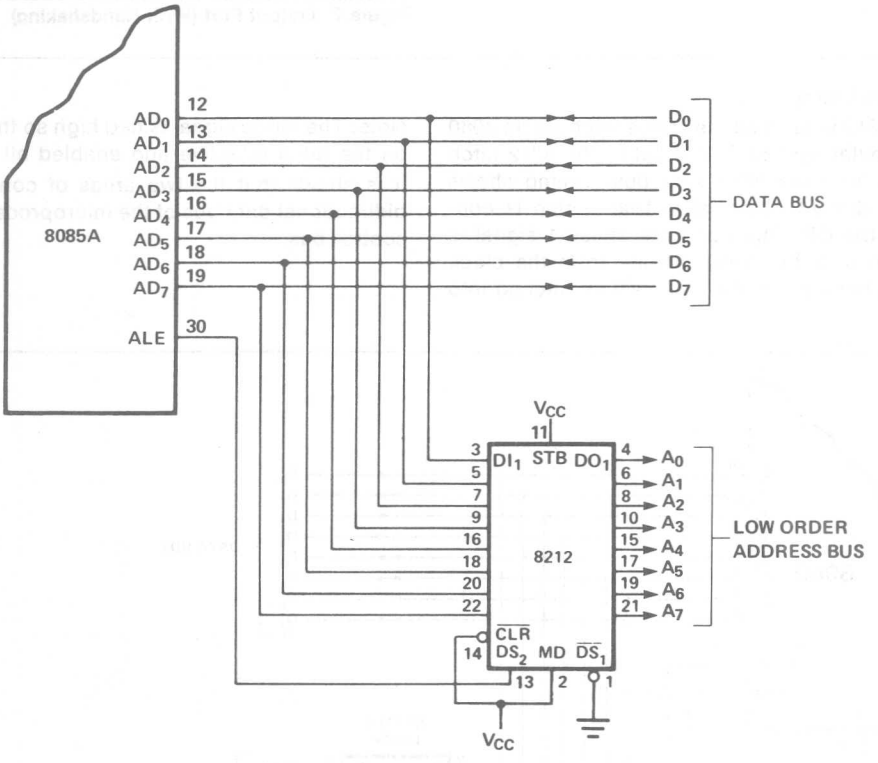


Figure 9. 8085A Low-Order Address Latch

ABSOLUTE MAXIMUM RATINGS*

Temperature under bias plastic 0°C to 75°C
Storage temperature 0°C to 75°C
All output or supply voltages – 0.5V to + 7V
All input voltages – 1.0V to + 5.5V
Output currents 100 mA

COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied.

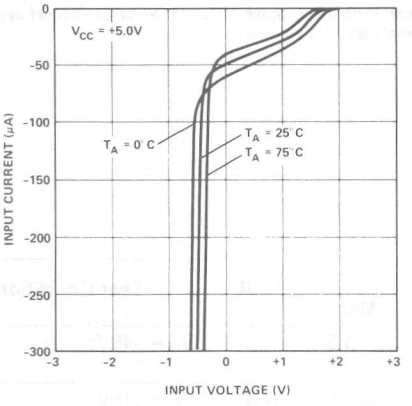
D.C. CHARACTERISTICS

T_A = 0°C to +75°C V_{CC} = +5V ±5%

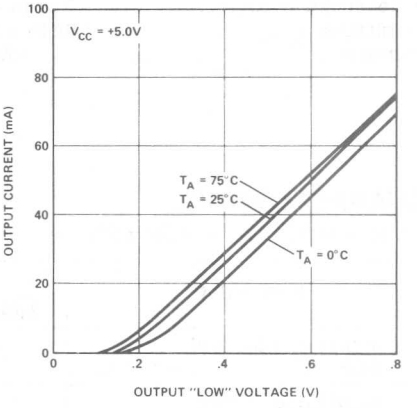
Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
I_F	Input Load Current ACK, DS_2, CR, DI_1-DI_8 Inputs			–.25	mA	V_F = .45V
I_F	Input Load Current MD Input			–.75	mA	V_F = .45V
I_F	Input Load Current DS_1 Input			–1.0	mA	V_F = .45V
I_R	Input Leakage Current ACK, DS, CR, DI_1-DI_8 Inputs			10	μA	$V_R \leq V_{CC}$
I_R	Input Leakage Current MO Input			30	μA	$V_R \leq V_{CC}$
I_R	Input Leakage Current DS Input			40	μA	$V_R \leq V_{CC}$
V_C	Input Forward Voltage Clamp			–1	V	I_C = –5 mA
V_{IL}	Input "Low" Voltage			.85	V	
V_{IH}	Input "High" Voltage	2.0			V	
V_{OL}	Output "Low" Voltage			.45	V	I_{OL} = 15 mA
V_{OH}	Output "High" Voltage	3.65	4.0		V	I_{OH} = –1 mA
I_{SC}	Short Circuit Output Current	–15		–75	mA	V_O = 0V, V_{CC} = 5.0V
I_O	Output Leakage Current High Impedance State			20	μA	V_O = .45V/5.25V
I_{CC}	Power Supply Current		90	130	mA	

TYPICAL CHARACTERISTICS

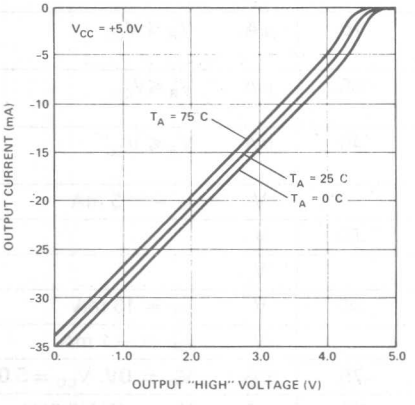
INPUT CURRENT VS. INPUT VOLTAGE



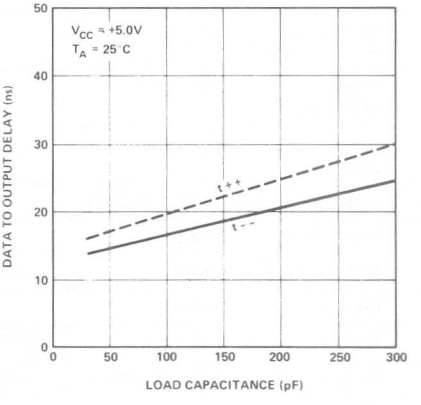
OUTPUT CURRENT VS. OUTPUT "LOW" VOLTAGE



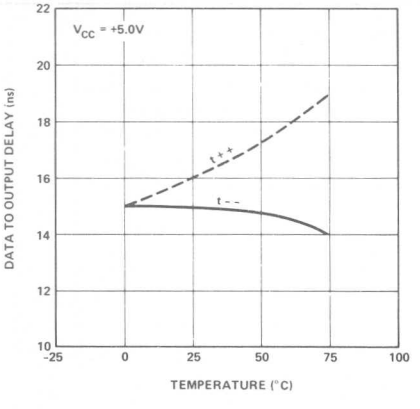
OUTPUT CURRENT VS. OUTPUT "HIGH" VOLTAGE



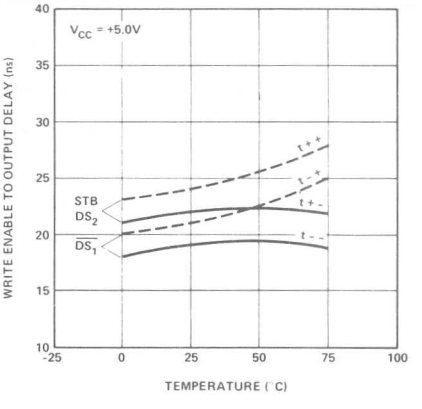
DATA TO OUTPUT DELAY VS. LOAD CAPACITANCE



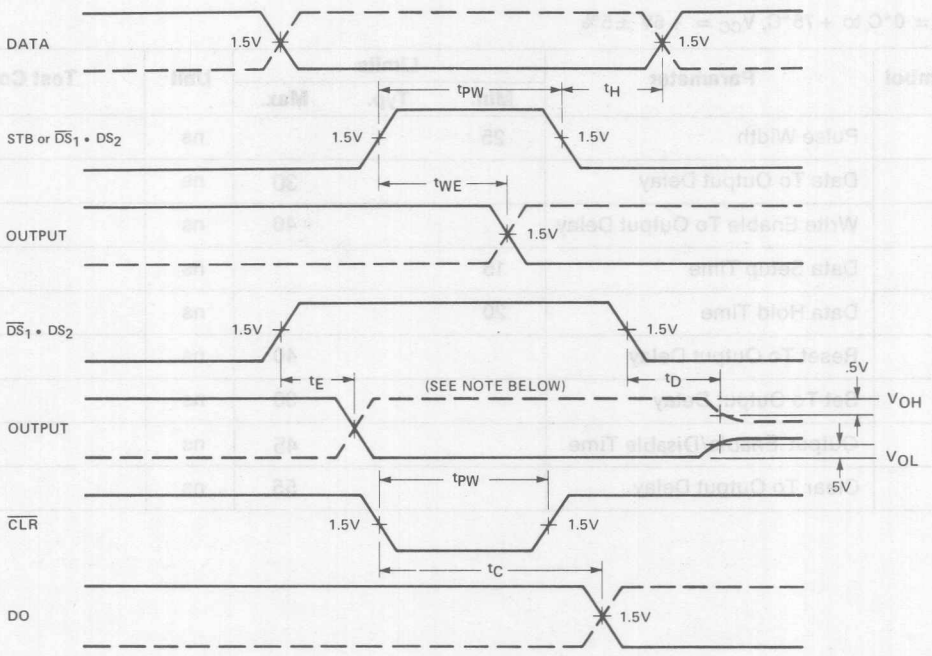
DATA TO OUTPUT DELAY VS. TEMPERATURE

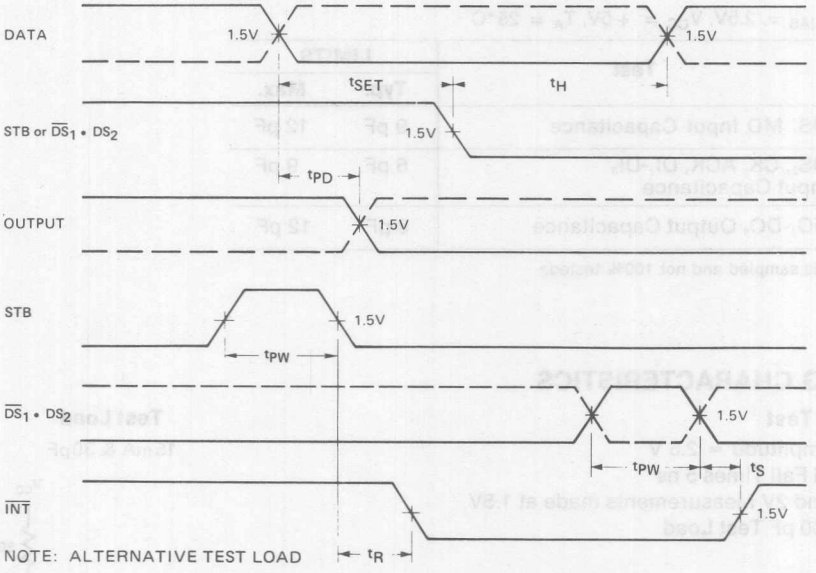


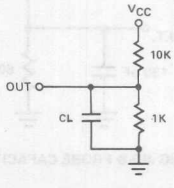
WRITE ENABLE TO OUTPUT DELAY VS. TEMPERATURE



TIMING DIAGRAM







A.C. CHARACTERISTICS

T_A = 0°C to +75°C, V_{CC} = +5V ±5%

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
t_{pw}	Pulse Width	25			ns	
t_{pd}	Data To Output Delay			30	ns	
t_{we}	Write Enable To Output Delay			40	ns	
t_{set}	Data Setup Time	15			ns	
t_h	Data Hold Time	20			ns	
t_r	Reset To Output Delay			40	ns	
t_s	Set To Output Delay			30	ns	
t_e	Output Enable/Disable Time			45	ns	
t_c	Clear To Output Delay			55	ns	

CAPACITANCE*

F = 1 MHz, V_{BIAS} = 2.5V, V_{CC} = +5V, T_A = 25°C

Symbol	Test	LIMITS	
		Typ.	Max.
C_{IN}	DS_1 MD Input Capacitance	9 pF	12 pF
C_{IN}	DS_2, CK, ACK, DI_1-DI_8 Input Capacitance	5 pF	9 pF
C_{OUT}	DO_1-DO_8 Output Capacitance	8 pF	12 pF

*This parameter is sampled and not 100% tested.

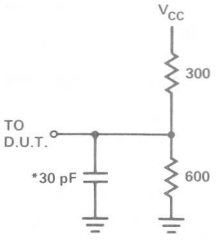
SWITCHING CHARACTERISTICS

Conditions of Test

Input Pulse Amplitude = 2.5 V
Input Rise and Fall Times 5 ns
Between 1V and 2V Measurements made at 1.5V
with 15 mA & 30 pF Test Load

Test Load

15mA & 30pF



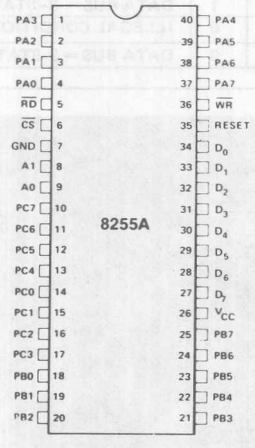
* INCLUDING JIG & PROBE CAPACITANCE

8255A/8255A-5
PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel® Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- 40-Pin Dual In-Line Package
- Reduces System Package Count
- Improved DC Driving Capability

The Intel® 8255A is a general purpose programmable I/O device designed for use with Intel® microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

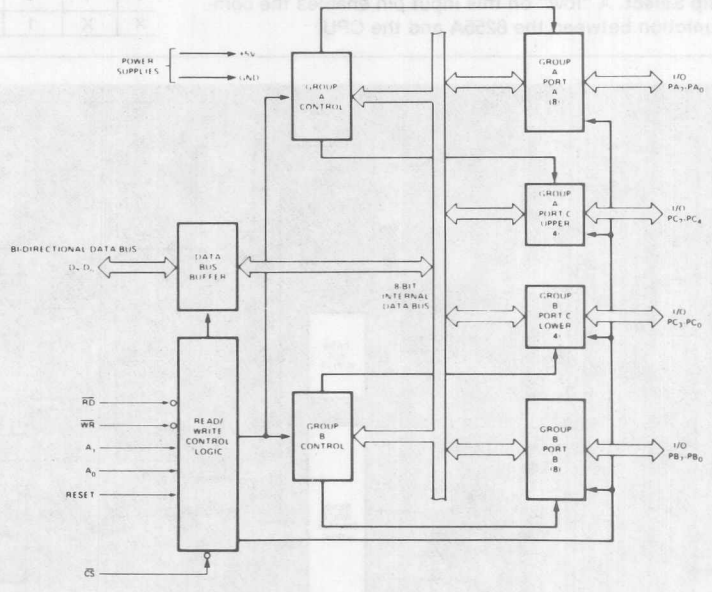
PIN CONFIGURATION



PIN NAMES

D_7–D_0	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
$\overline{CS}$	CHIP SELECT
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
V_{CC}	+5 VOLTS
GND	0 VOLTS

8255A BLOCK DIAGRAM



© Intel Corporation, 1978

February, 1978

8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel® microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

($\overline{CS}$)

Chip Select. A "low" on this input pin enables the communiction between the 8255A and the CPU.

($\overline{RD}$)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

($\overline{WR}$)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A_0 and A_1)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

8255A BASIC OPERATION

A_1	A_0	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0	0	0	1	0	PORT A ⇒ DATA BUS
0	1	0	1	0	PORT B ⇒ DATA BUS
1	0	0	1	0	PORT C ⇒ DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS ⇒ PORT A
0	1	1	0	0	DATA BUS ⇒ PORT B
1	0	1	0	0	DATA BUS ⇒ PORT C
1	1	1	0	0	DATA BUS ⇒ CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS ⇒ 3-STATE
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS ⇒ 3-STATE

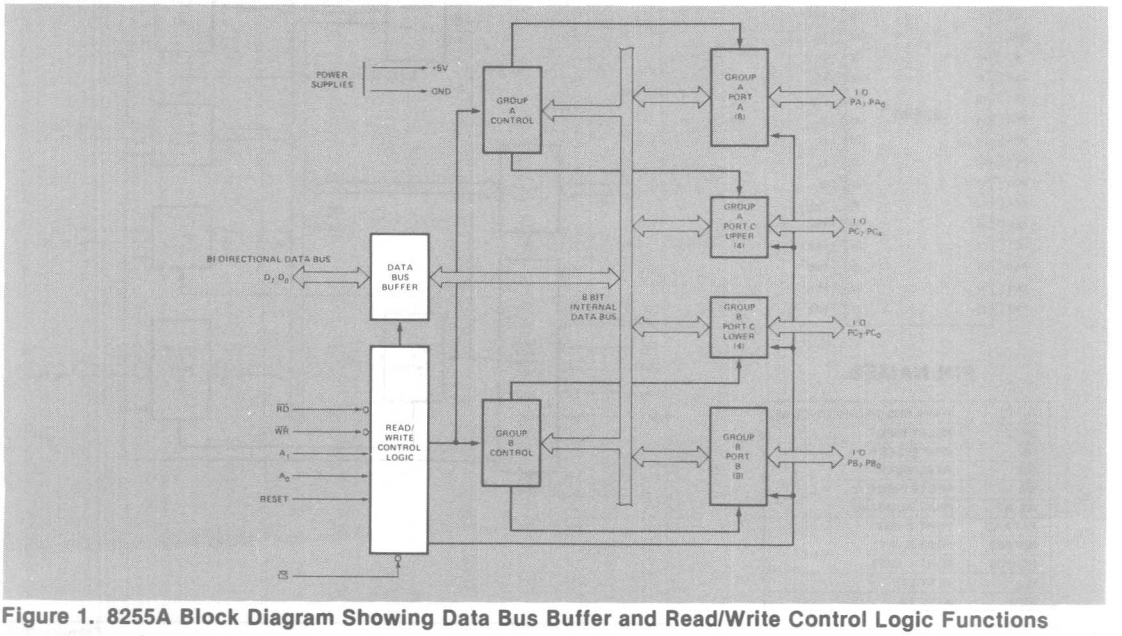


Figure 1. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions

(RESET)

Reset. A "high on this input clears the control register and all ports (A, C, C) are set to the input mode.

Group A and Group B Controls

The functional configuration of each port is programmed by the systems software. In essence, the CPU "outputs" a control word to the 8255A. The control word contains information such as "mode", "bit set", "bit reset", etc., that initializes the functional configuration of the 8255.

Each of the Control blocks (Group A and Group B) accepts "commands" from the Read/Write Control Logic, receives "control words" from the internal data bus and issues the proper commands to its associated ports.

Control Group A – Port A and Port C upper (C7-C4)
Control Group B – Port B and Port C lower (C3-C0)

The Control Word Register can **Only** be written into. No Read operation of the Control Word Register is allowed.

Ports A, B, and C

The 8255A contains three 8-bit ports (A, B, and C). All can be configured in a wide variety of functional characteristics by the system software but each has its own special features or "personality" to further enhance the power and flexibility of the 8255A.

Port A. One 8-bit data output latch/buffer and one 8-bit data input latch.

Port B. One 8-bit data input/output latch/buffer and one 8-bit data input buffer.

Port C. One 8-bit data output latch/buffer and one 8-bit data input buffer (no latch for input). This port can be divided into two 4-bit ports under the mode control. Each 4-bit port contains a 4-bit latch and it can be used for the control signal outputs and status signal inputs in conjunction with ports A and B.

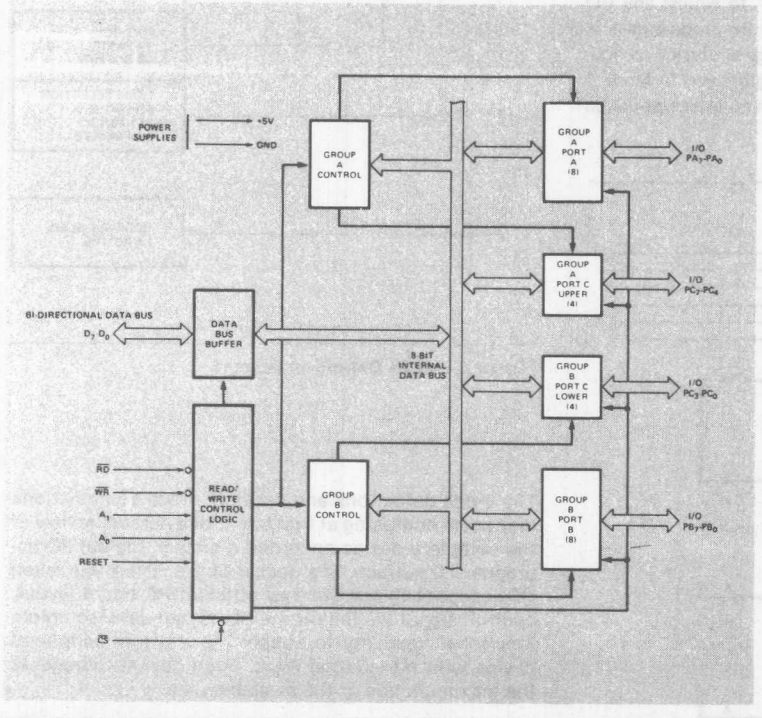
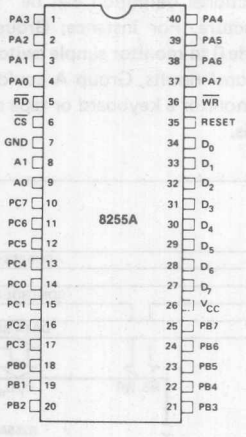


Figure 2. 8225A Block Diagram Showing Group A and Group B Control Functions

PIN CONFIGURATION



PIN NAMES

D_7–D_0	DATA BUS (BI-DIRECTIONAL)
RESET	RESET INPUT
$\overline{CS}$	CHIP SELECT
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A0, A1	PORT ADDRESS
PA7-PA0	PORT A (BIT)
PB7-PB0	PORT B (BIT)
PC7-PC0	PORT C (BIT)
V_{CC}	+5 VOLTS
GND	0 VOLTS

8255A OPERATIONAL DESCRIPTION

Mode Selection

There are three basic modes of operation that can be selected by the system software:

Mode 0 — Basic Input/Output
Mode 1 — Strobed Input/Output
Mode 2 — Bi-Directional Bus

When the reset input goes "high" all ports will be set to the input mode (i.e., all 24 lines will be in the high impedance state). After the reset is removed the 8255A can remain in the input mode with no additional initialization required. During the execution of the system program any of the other modes may be selected using a single output instruction. This allows a single 8255A to service a variety of peripheral devices with a simple software maintenance routine.

The modes for Port A and Port B can be separately defined, while Port C is divided into two portions as required by the Port A and Port B definitions. All of the output registers, including the status flip-flops, will be reset whenever the mode is changed. Modes may be combined so that their functional definition can be "tailored" to almost any I/O structure. For instance; Group B can be programmed in Mode 0 to monitor simple switch closings or display computational results, Group A could be programmed in Mode 1 to monitor a keyboard or tape reader on an interrupt-driven basis.

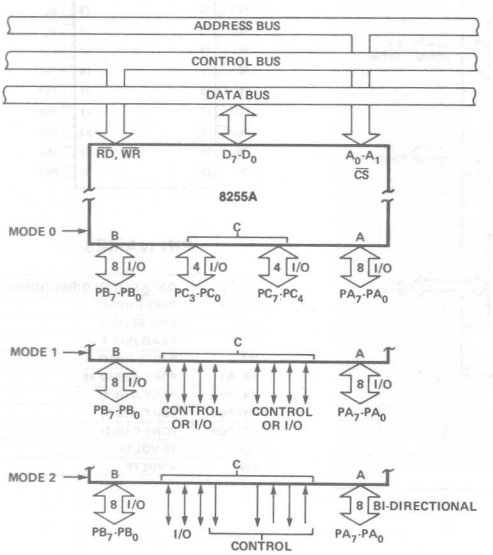


Figure 3. Basic Mode Definitions and Bus Interface

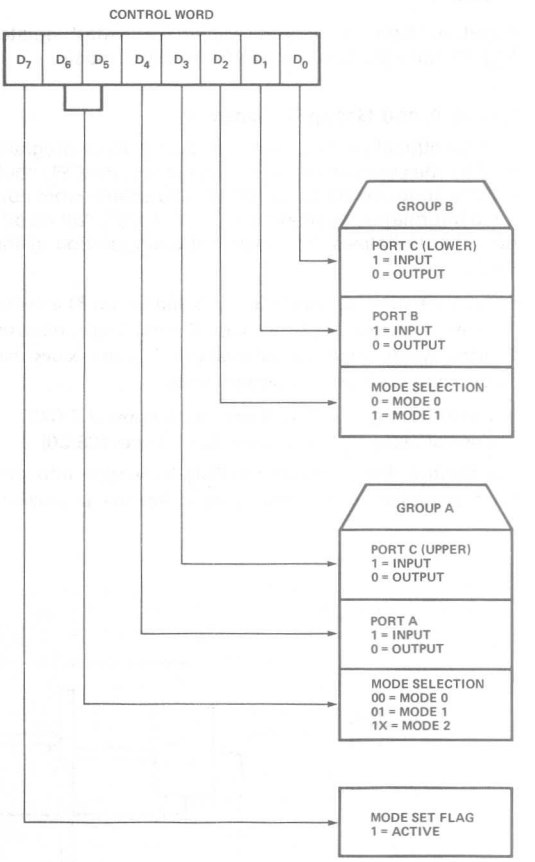


Figure 4. Mode Definition Format

The mode definitions and possible mode combinations may seem confusing at first but after a cursory review of the complete device operation a simple, logical I/O approach will surface. The design of the 8255A has taken into account things such as efficient PC board layout, control signal definition vs PC layout and complete functional flexibility to support almost any peripheral device with no external logic. Such design represents the maximum use of the available pins.

Single Bit Set/Reset Feature

Any of the eight bits of Port C can be Set or Reset using a single OUTput instruction. This feature reduces software requirements in Control-based applications.

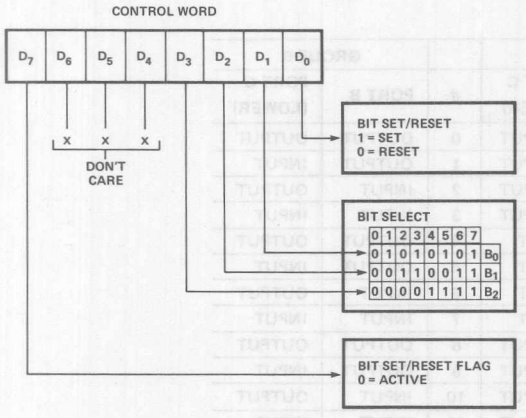


Figure 5. Bit Set/Reset Format

When Port C is being used as status/control for Port A or B, these bits can be set or reset by using the Bit Set/Reset operation just as if they were data output ports.

Interrupt Control Functions

When the 8255A is programmed to operate in mode 1 or mode 2, control signals are provided that can be used as interrupt request inputs to the CPU. The interrupt request signals, generated from port C, can be inhibited or enabled by setting or resetting the associated INTE flip-flop, using the bit set/reset function of port C.

This function allows the Programmer to disallow or allow a specific I/O device to interrupt the CPU without affecting any other device in the interrupt structure.

INTE flip-flop definition:

(BIT-SET) — INTE is SET — Interrupt enable
(BIT-RESET) — INTE is RESET — Interrupt disable

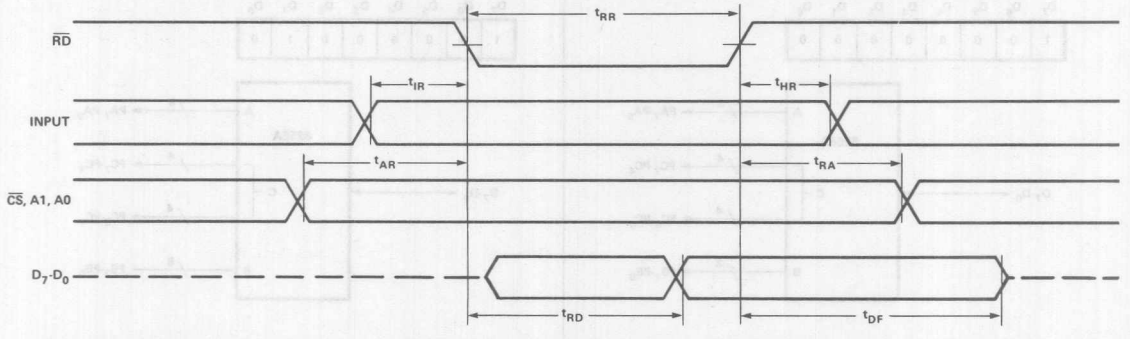
Note: All Mask flip-flops are automatically reset during mode selection and device Reset.

Operating Modes

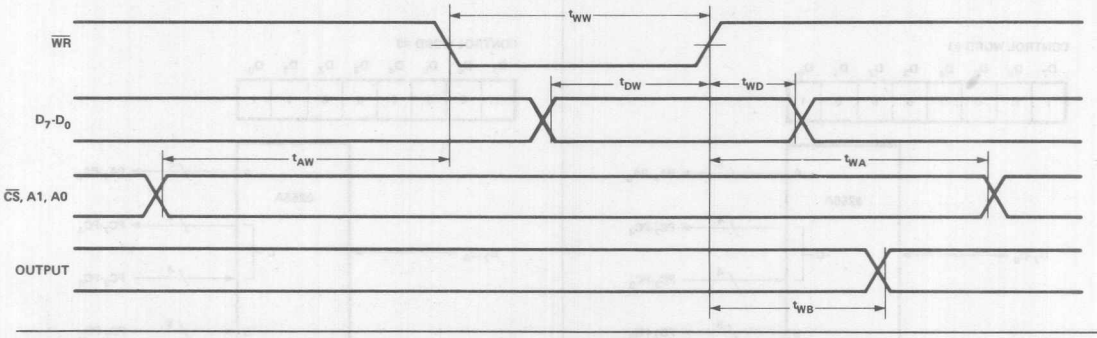
MODE 0 (Basic Input/Output). This functional configuration provides simple input and output operations for each of the three ports. No "handshaking" is required, data is simply written to or read from a specified port.

Mode 0 Basic Functional Definitions:

- Two 8-bit ports and two 4-bit ports.
- Any port can be input or output.
- Outputs are latched.
- Inputs are not latched.
- 16 different Input/Output configurations are possible in this Mode.



MODE 0 (Basic Input)



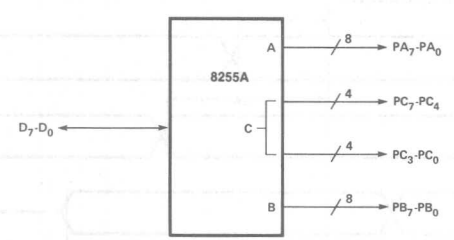
MODE 0 (Basic Output)

A		B		GROUP A			GROUP B	
D_4	D_3	D_1	D_0	PORT A	PORT C (UPPER)	#	PORT B	PORT C (LOWER)
0	0	0	0	OUTPUT	OUTPUT	0	OUTPUT	OUTPUT
0	0	0	1	OUTPUT	OUTPUT	1	OUTPUT	INPUT
0	0	1	0	OUTPUT	OUTPUT	2	INPUT	OUTPUT
0	0	1	1	OUTPUT	OUTPUT	3	INPUT	INPUT
0	1	0	0	OUTPUT	INPUT	4	OUTPUT	OUTPUT
0	1	0	1	OUTPUT	INPUT	5	OUTPUT	INPUT
0	1	1	0	OUTPUT	INPUT	6	INPUT	OUTPUT
0	1	1	1	OUTPUT	INPUT	7	INPUT	INPUT
1	0	0	0	INPUT	OUTPUT	8	OUTPUT	OUTPUT
1	0	0	1	INPUT	OUTPUT	9	OUTPUT	INPUT
1	0	1	0	INPUT	OUTPUT	10	INPUT	OUTPUT
1	0	1	1	INPUT	OUTPUT	11	INPUT	INPUT
1	1	0	0	INPUT	INPUT	12	OUTPUT	OUTPUT
1	1	0	1	INPUT	INPUT	13	OUTPUT	INPUT
1	1	1	0	INPUT	INPUT	14	INPUT	OUTPUT
1	1	1	1	INPUT	INPUT	15	INPUT	INPUT

MODE 0 Configurations

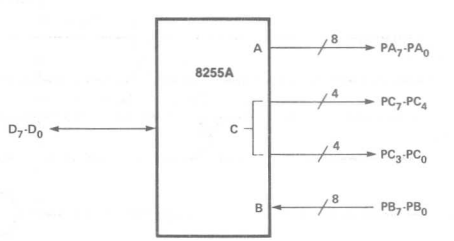
CONTROL WORD #0

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	0



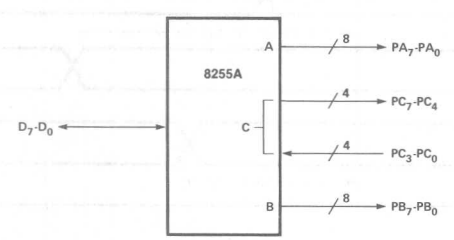
CONTROL WORD #2

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	0



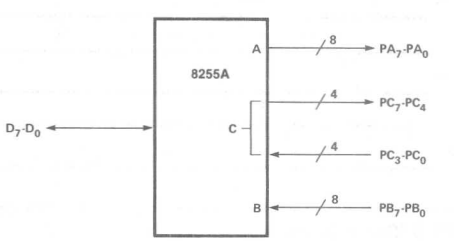
CONTROL WORD #1

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	0	1



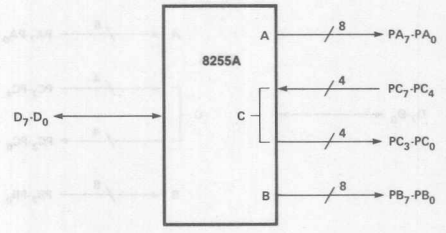
CONTROL WORD #3

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	0	0	1	1



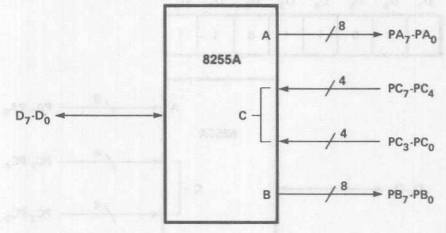
CONTROL WORD #4

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	0



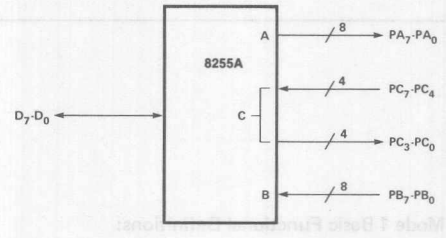
CONTROL WORD #5

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	0	1



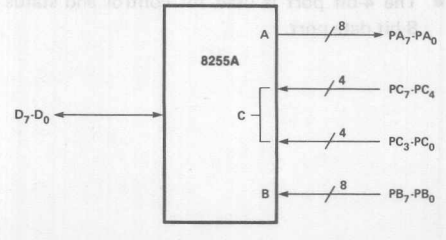
CONTROL WORD #6

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	0



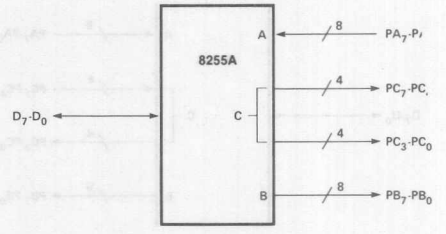
CONTROL WORD #7

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	0	1	0	1	1



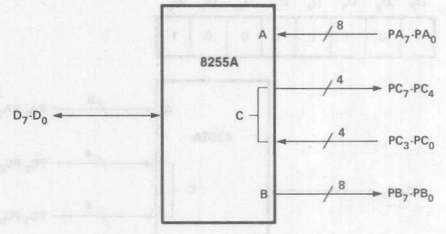
CONTROL WORD #8

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	0



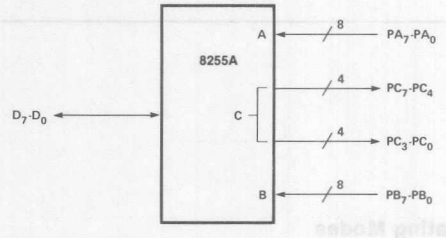
CONTROL WORD #9

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	0	1



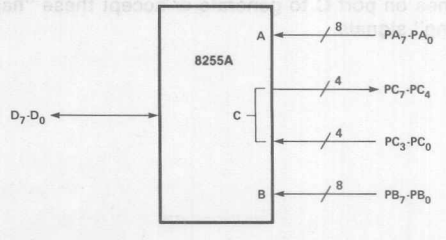
CONTROL WORD #10

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	0



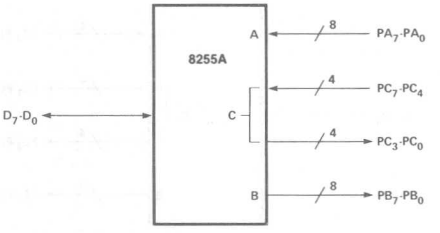
CONTROL WORD #11

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	0	0	1	1



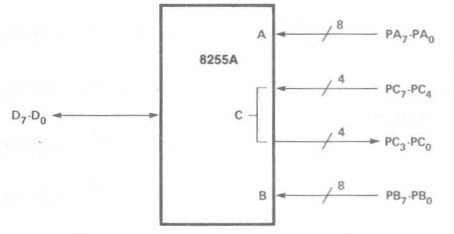
CONTROL WORD #12

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	0	0



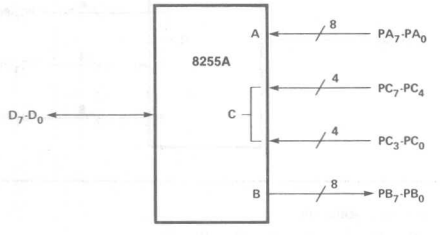
CONTROL WORD #14

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	1	0



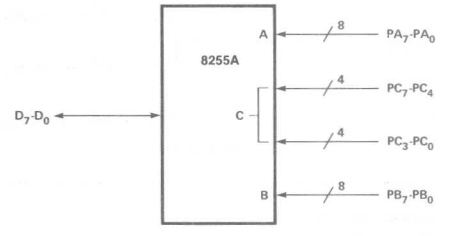
CONTROL WORD #13

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	0	1



CONTROL WORD #15

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
1	0	0	1	1	0	1	1



Operating Modes

MODE 1 (Strobed Input/Output). This functional configuration provides a means for transferring I/O data to or from a specified port in conjunction with strobes or "handshaking" signals. In mode 1, port A and Port B use the lines on port C to generate or accept these "handshaking" signals.

Mode 1 Basic Functional Definitions:

- Two Groups (Group A and Group B)
- Each group contains one 8-bit data port and one 4-bit control/data port.
- The 8-bit data port can be either input or output. Both inputs and outputs are latched.
- The 4-bit port is used for control and status of the 8-bit data port.

Input Control Signal Definition

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F)

A "high" on this output indicates that the data has been loaded into the input latch; in essence, an acknowledgement. IBF is set by STB input being low and is reset by the rising edge of the RD input.

INTR (Interrupt Request)

A "high" on this output can be used to interrupt the CPU when an input device is requesting service. INTR is set by the $\overline{STB}$ is a "one", IBF is a "one" and INTE is a "one". It is reset by the falling edge of $\overline{RD}$. This procedure allows an input device to request service from the CPU by simply strobing its data into the port.

INTE A

Controlled by bit set/reset of PC_4.

INTE B

Controlled by bit set/reset of PC_2.

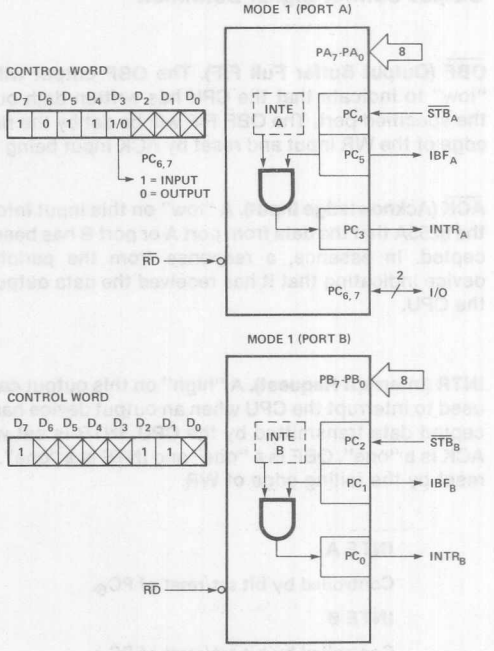


Figure 6. MODE 1 Input

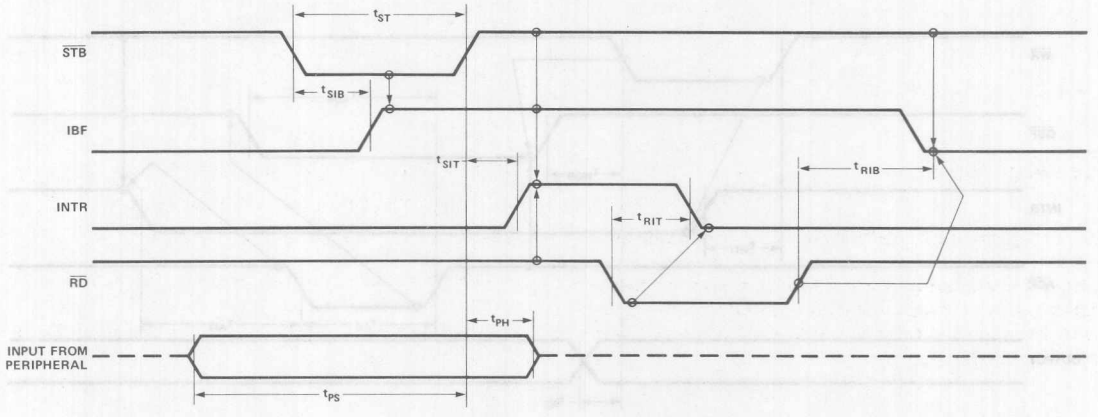


Figure 7. MODE 1 (Strobed Input)

Output Control Signal Definition

$\overline{OBF}$ (Output Buffer Full F/F). The OBF output will go "low" to indicate that the CPU has written data out to the specified port. The OBF F/F will be set by the rising edge of the WR input and reset by ACK Input being low.

$\overline{ACK}$ (Acknowledge Input). A "low" on this input informs the 8255A that the data from port A or port B has been accepted. In essence, a response from the peripheral device indicating that it has received the data output by the CPU.

INTR (Interrupt Request). A "high" on this output can be used to interrupt the CPU when an output device has accepted data transmitted by the CPU. INTR is set when ACK is a "one", OBF is a "one" and INTE is a "one". It is reset by the falling edge of WR.

INTE A

Controlled by bit set/reset of PC_6.

INTE B

Controlled by bit set/reset of PC_2.

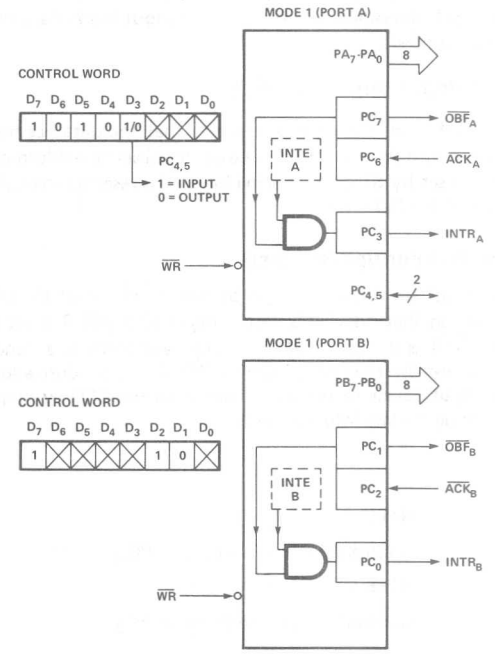


Figure 8. MODE 1 Output

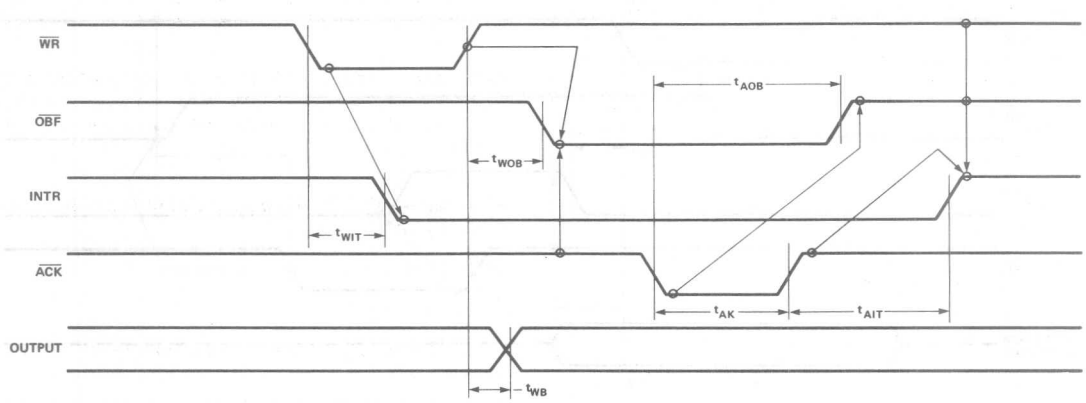


Figure 9. Mode 1 (Strobed Output)

Combinations of MODE 1

Port A and Port B can be individually defined as input or output in Mode 1 to support a wide variety of strobed I/O applications.

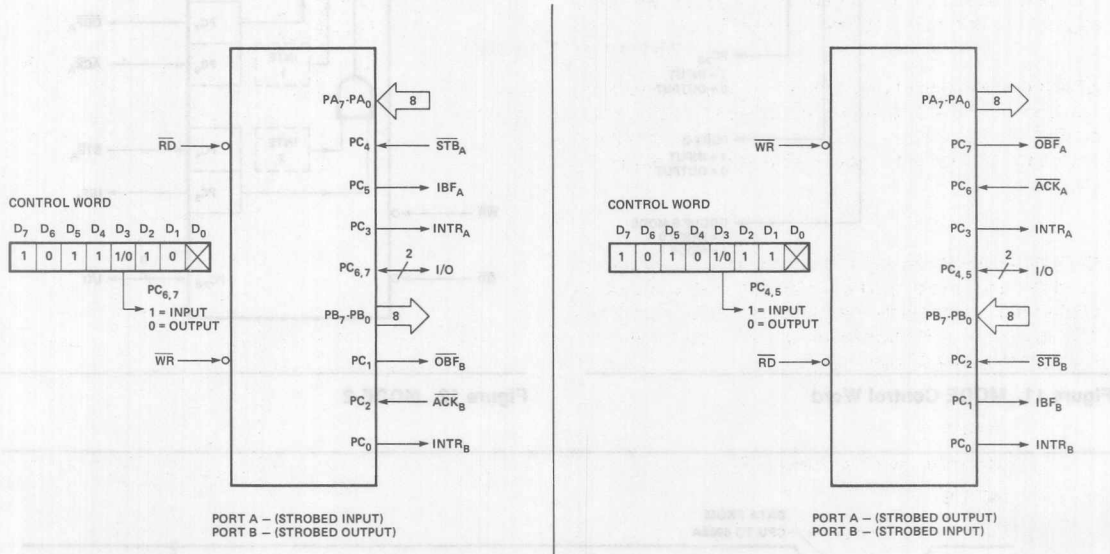


Figure 10. Combinations of MODE 1

Operating Modes

MODE 2 (Strobed Bidirectional Bus I/O). This functional configuration provides a means for communicating with a peripheral device or structure on a single 8-bit bus for both transmitting and receiving data (bidirectional bus I/O). "Handshaking" signals are provided to maintain proper bus flow discipline in a similar manner to MODE 1. Interrupt generation and enable/disable functions are also available.

MODE 2 Basic Functional Definitions:

- Used in Group A only.
- One 8-bit, bi-directional bus Port (Port A) and a 5-bit control Port (Port C).
- Both inputs and outputs are latched.
- The 5-bit control port (Port C) is used for control and status for the 8-bit, bi-directional bus port (Port A).

Bidirectional Bus I/O Control Signal Definition

INTR (Interrupt Request). A high on this output can be used to interrupt the CPU for both input or output operations.

Output Operations

$\overline{\text{OBF}}$ (Output Buffer Ful). The OBF output will go "low" to indicate that the CPU has written data out to port A.

$\overline{\text{ACK}}$ (Acknowledge). A "low" on this input enables the tri-state output buffer of port A to send out the data. Otherwise, the output buffer will be in the high impedance state.

INTE 1 (The INTE Flip-Flop Associated with OBF). Controlled by bit set/reset of PC_6.

Input Operations

$\overline{\text{STB}}$ (Strobe Input)

STB (Strobe Input). A "low" on this input loads data into the input latch.

IBF (Input Buffer Full F/F). A "high" on this output indicates that data has been loaded into the input latch.

INTE 2 (The INTE Flip-Flop Associated with IBF). Controlled by bit set/reset of PC_4.

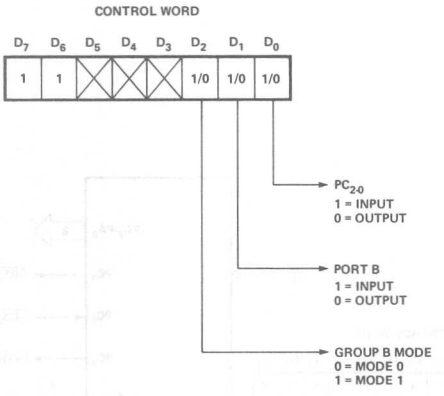


Figure 11. MODE Control Word

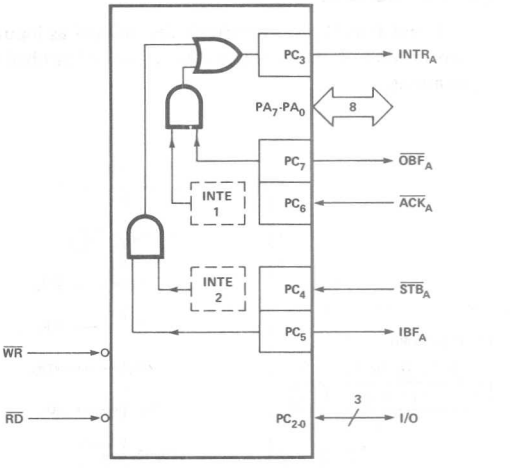


Figure 12. MODE 2

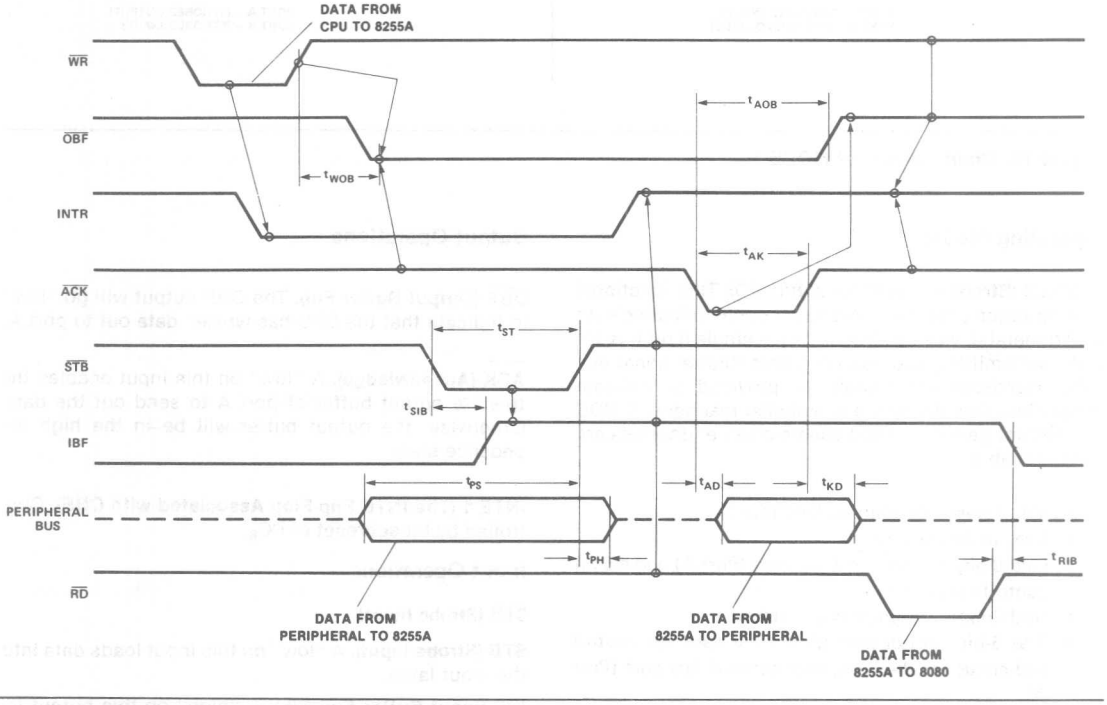


Figure 13. MODE 2 (Bidirectional)

NOTE: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

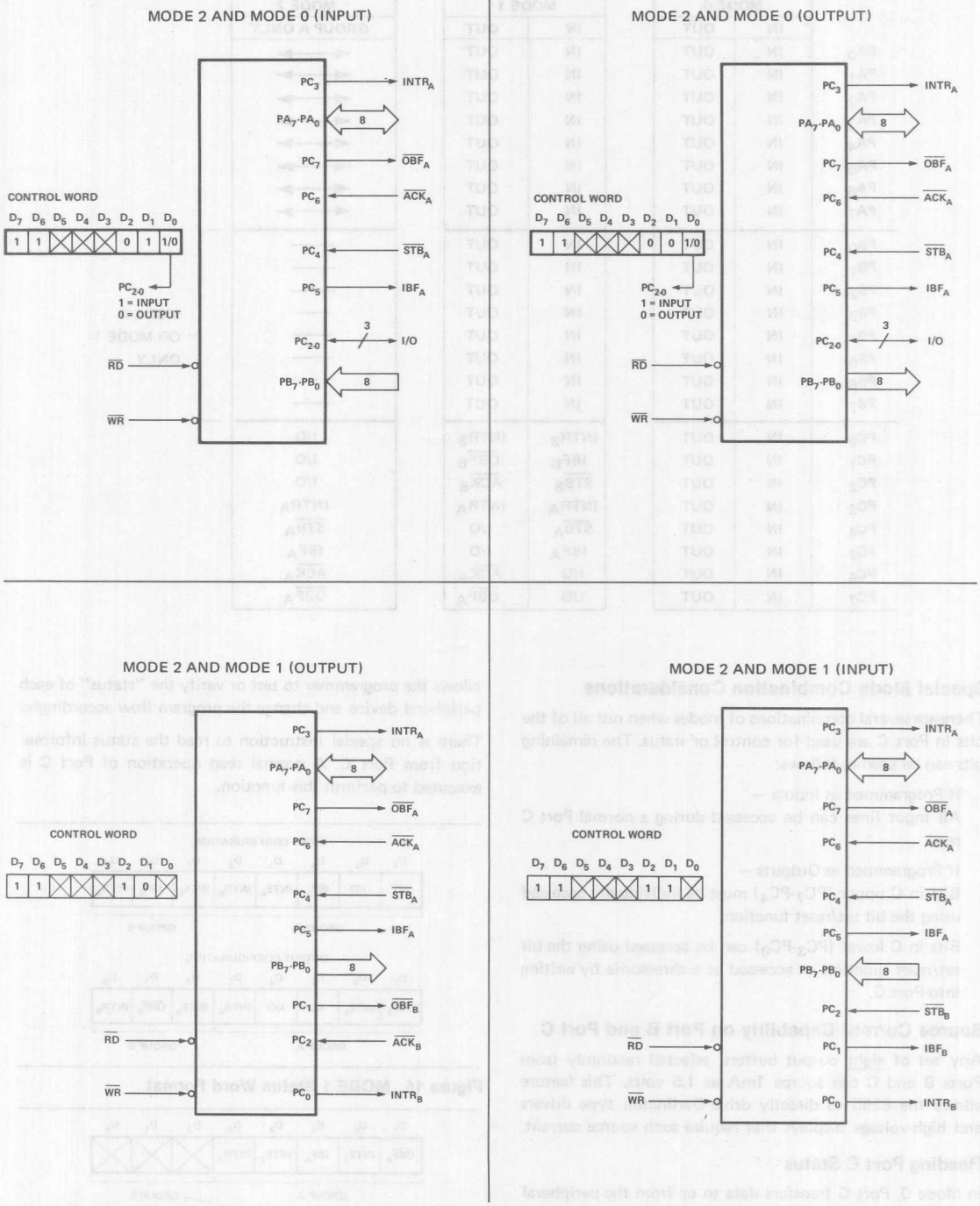


Figure 14. MODE 2 Combinations

	MODE 0		MODE 1		MODE 2	
	IN	OUT	IN	OUT	GROUP A ONLY	
PA_0	IN	OUT	IN	OUT	↔	
PA_1	IN	OUT	IN	OUT	↔	
PA_2	IN	OUT	IN	OUT	↔	
PA_3	IN	OUT	IN	OUT	↔	
PA_4	IN	OUT	IN	OUT	↔	
PA_5	IN	OUT	IN	OUT	↔	
PA_6	IN	OUT	IN	OUT	↔	
PA_7	IN	OUT	IN	OUT	↔	
PB_0	IN	OUT	IN	OUT	—	MODE 0 OR MODE 1 ONLY
PB_1	IN	OUT	IN	OUT	—	
PB_2	IN	OUT	IN	OUT	—	
PB_3	IN	OUT	IN	OUT	—	
PB_4	IN	OUT	IN	OUT	—	
PB_5	IN	OUT	IN	OUT	—	
PB_6	IN	OUT	IN	OUT	—	
PB_7	IN	OUT	IN	OUT	—	
PC_0	IN	OUT	$INTR_B$	$INTR_B$	I/O	
PC_1	IN	OUT	IBF_B	$\overline{OBF}_B$	I/O	
PC_2	IN	OUT	$\overline{STB}_B$	$\overline{ACK}_B$	I/O	
PC_3	IN	OUT	$INTR_A$	$INTR_A$	$INTR_A$	
PC_4	IN	OUT	$\overline{STB}_A$	I/O	$\overline{STB}_A$	
PC_5	IN	OUT	IBF_A	I/O	IBF_A	
PC_6	IN	OUT	I/O	$\overline{ACK}_A$	$\overline{ACK}_A$	
PC_7	IN	OUT	I/O	$\overline{OBF}_A$	$\overline{OBF}_A$	

Special Mode Combination Considerations

There are several combinations of modes when not all of the bits in Port C are used for control or status. The remaining bits can be used as follows:

If Programmed as Inputs –
All input lines can be accessed during a normal Port C read.

If Programmed as Outputs –
Bits in C upper (PC_7-PC_4) must be individually accessed using the bit set/reset function.

Bits in C lower (PC_3-PC_0) can be accessed using the bit set/reset function or accessed as a threesome by writing into Port C.

Source Current Capability on Port B and Port C

Any set of **eight** output buffers, selected randomly from Ports B and C can source 1mA at 1.5 volts. This feature allows the 8255 to directly drive Darlington type drivers and high-voltage displays that require such source current.

Reading Port C Status

In Mode 0, Port C transfers data to or from the peripheral device. When the 8255 is programmed to function in Modes 1 or 2, Port C generates or accepts "hand-shaking" signals with the peripheral device. Reading the contents of Port C allows the programmer to test or verify the "status" of each peripheral device and change the program flow accordingly.

There is no special instruction to read the status information from Port C. A normal read operation of Port C is executed to perform this function.

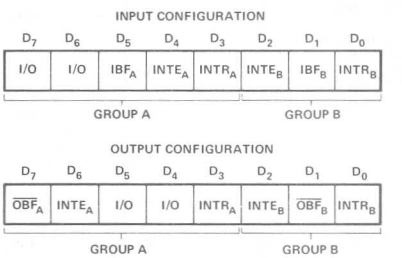


Figure 15. MODE 1 Status Word Format

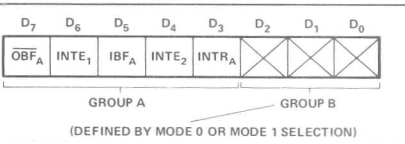


Figure 16. MODE 2 Status Word Format

APPLICATIONS OF THE 8255A

The 8255A is a very powerful tool for interfacing peripheral equipment to the microcomputer system. It represents the optimum use of available pins and is flexible enough to interface almost any I/O device without the need for additional external logic.

Each peripheral device in a microcomputer system usually has a "service routine" associated with it. The routine manages the software interface between the device and the CPU. The functional definition of the 8255A is programmed by the I/O service routine and becomes an extension of the system software. By examining the I/O devices interface characteristics for both data transfer and timing, and matching this information to the examples and tables in the detailed operational description, a control word can easily be developed to initialize the 8255A to exactly "fit" the application. Figures 17 through 23 present a few examples of typical applications of the 8255A.

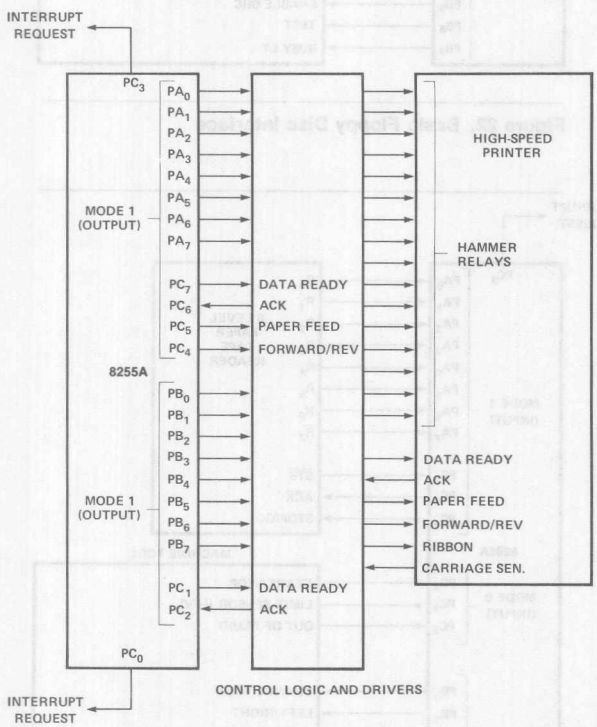


Figure 17. Printer Interface

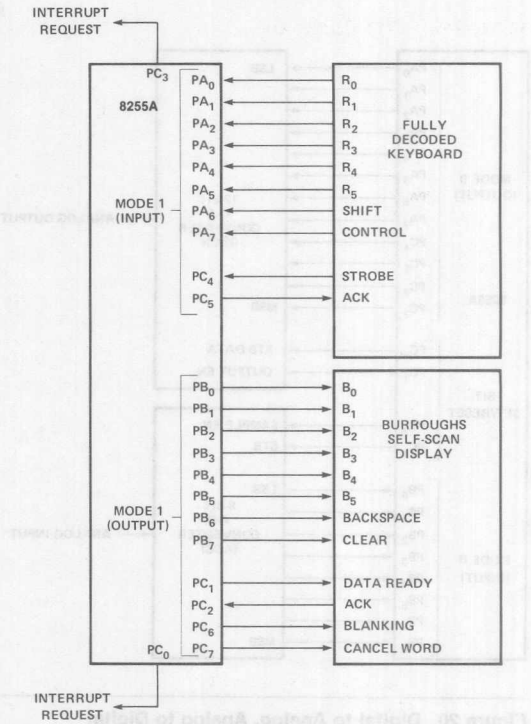


Figure 18. Keyboard and Display Interface

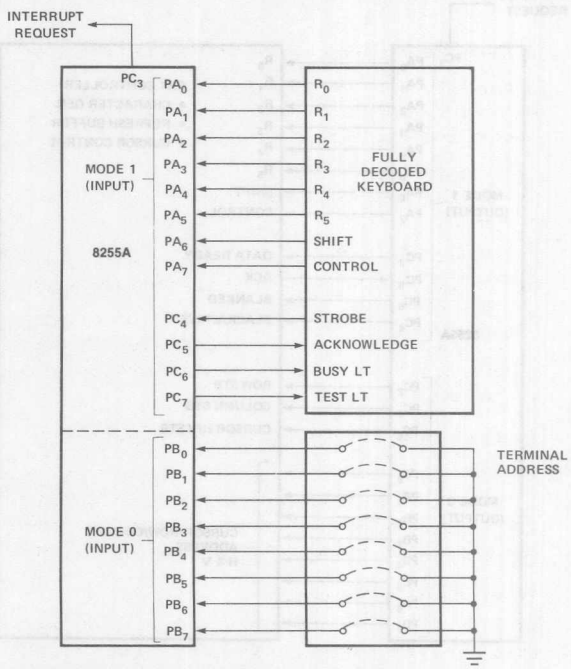


Figure 19. Keyboard and Terminal Address Interface

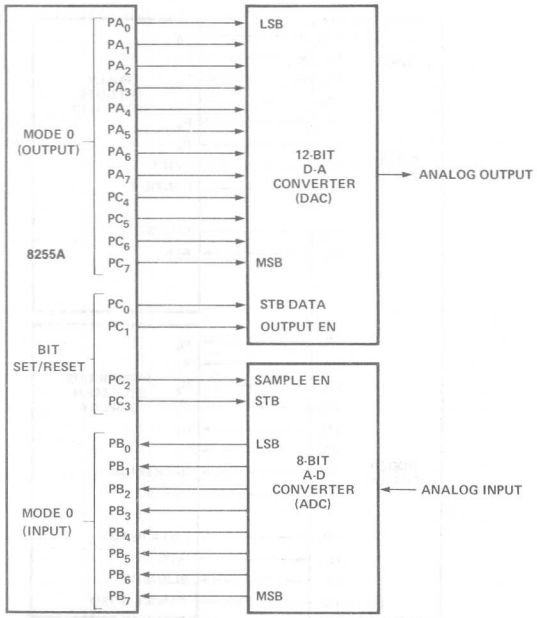


Figure 20. Digital to Analog, Analog to Digital

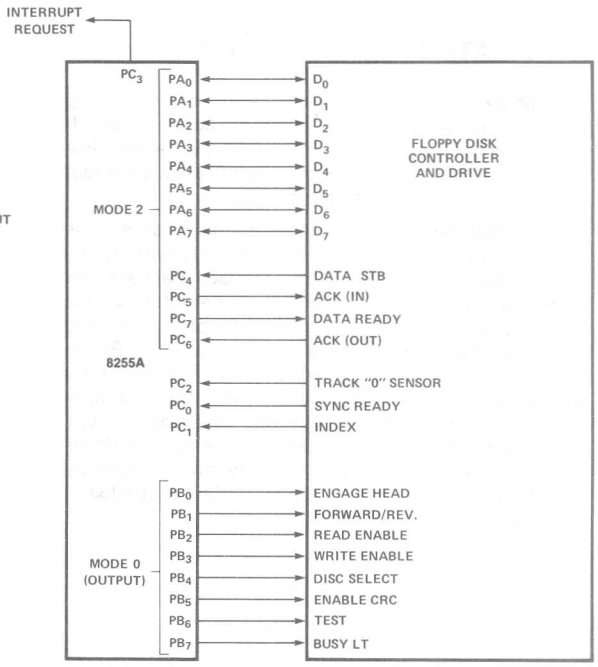


Figure 22. Basic Floppy Disc Interface

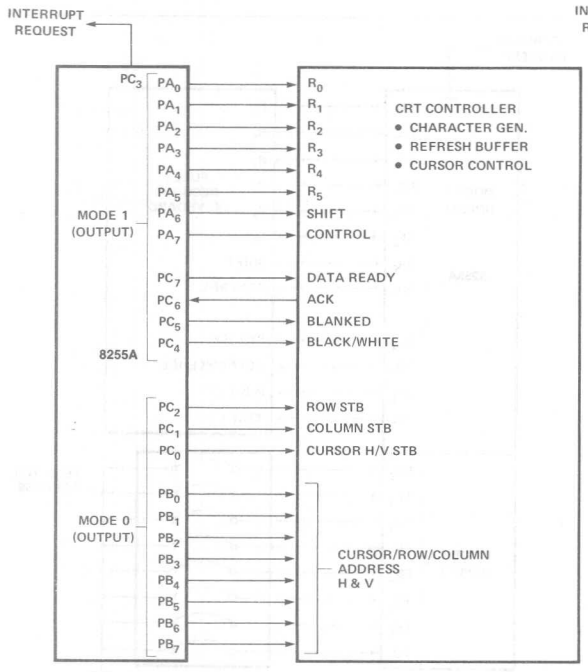


Figure 21. Basic CRT Controller Interface

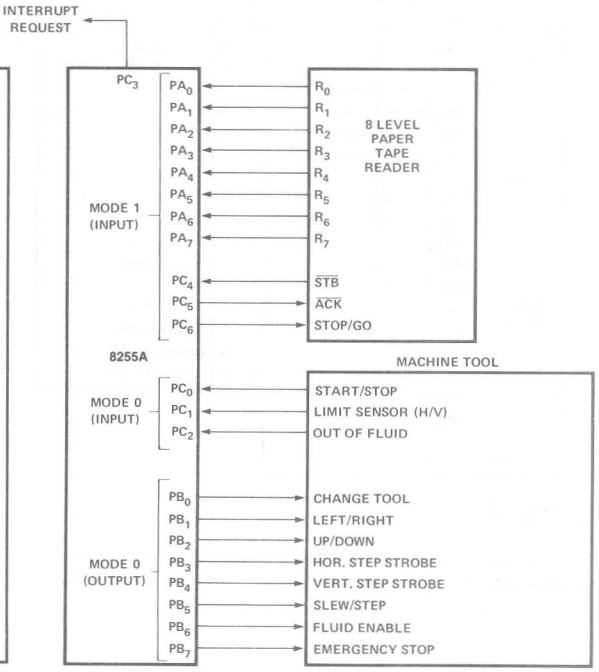


Figure 23. Machine Tool Controller Interface

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias. 0°C to 70°C
Storage Temperature –65°C to +150°C
Voltage on Any Pin
With Respect to Ground –0.5V to +7V
Power Dissipation 1 Watt

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = +5V ±5%; GND = 0V

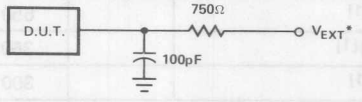
SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL} (DB)	Output Low Voltage (Data Bus)		0.45	V	I_{OL} = 2.5mA
V_{OL}(PER)	Output Low Voltage (Peripheral Port)		0.45	V	I_{OL} = 1.7mA
V_{OH}(DB)	Output High Voltage (Data Bus)	2.4		V	I_{OH} = -400μA
V_{OH}(PER)	Output High Voltage (Peripheral Port)	2.4		V	I_{OH} = -200μA
I_{DAR}[1]	Darlington Drive Current	-1.0	-4.0	mA	R_{EXT} = 750Ω; V_{EXT} = 1.5V
I_{CC}	Power Supply Current		120	mA	
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0V

Note 1: Available on any 8 pins from Port B and C.

CAPACITANCE

T_A = 25°C; V_{CC} = GND = 0V

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
C_{IN}	Input Capacitance			10	pF	fc = 1MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to GND



*V_{EXT} is set at various voltages during testing to guarantee the specification.

Figure 24. Test Load Circuit (for dB)

A.C. CHARACTERISTICS

T_A = 0°C to 70°C; V_{CC} = +5V ±5%; GND = 0V

NOTE: The 8255A-5 specifications are not final. Some parametric limits are subject to change.

Bus Parameters

Read:

SYMBOL	PARAMETER	8255A MIN.	8255A MAX.	8255A-5 MIN.	8255A-5 MAX.	UNIT
t_{AR}	Address Stable Before READ	0		0		ns
t_{RA}	Address Stable After READ	0		0		ns
t_{RR}	READ Pulse Width	300		300		ns
t_{RD}	Data Valid From READ[1]		250		200	ns
t_{DF}	Data Float After READ	10	150	10	100	ns
t_{RV}	Time Between READs and/or WRITEs	850		850		ns

Write:

SYMBOL	PARAMETER	8255A MIN.	8255A MAX.	8255A-5 MIN.	8255A-5 MAX.	UNIT
t_{AW}	Address Stable Before WRITE	0		0		ns
t_{WA}	Address Stable After WRITE	20		20		ns
t_{WW}	WRITE Pulse Width	400		300		ns
t_{DW}	Data Valid to WRITE (T.E.)	100		100		ns
t_{WD}	Data Valid After WRITE	30		30		ns

Other Timings:

SYMBOL	PARAMETER	8255A MIN.	8255A MAX.	8255A-5 MIN.	8255A-5 MAX.	UNIT
t_{WB}	WR = 1 to Output[1]		350		350	ns
t_{IR}	Peripheral Data Before RD	0		0		ns
t_{HR}	Peripheral Data After RD	0		0		ns
t_{AK}	ACK Pulse Width	300		300		ns
t_{ST}	STB Pulse Width	500		500		ns
t_{PS}	Per. Data Before T.E. of STB	0		0		ns
t_{PH}	Per. Data After T.E. of STB	180		180		ns
t_{AD}	ACK = 0 to Output[1]		300		300	ns
t_{KD}	ACK = 1 to Output Float	20	250	20	250	ns
t_{WOB}	WR = 1 to OBF = 0[1]		650		650	ns
t_{AOB}	ACK = 0 to OBF = 1[1]		350		350	ns
t_{SIB}	STB = 0 to IBF = 1[1]		300		300	ns
t_{RIB}	RD = 1 to IBF = 0[1]		300		300	ns
t_{RIT}	RD = 0 to INTR = 0[1]		400		400	ns
t_{SIT}	STB = 1 to INTR = 1[1]		300		300	ns
t_{AIT}	ACK = 1 to INTR = 1[1]		350		350	ns
t_{WIT}	WR = 0 to INTR = 0[1]		850		850	ns

Notes: 1. Test Conditions: 8255A: C_L = 100pF; 8255A-5: C_L = 150pF.
2. Period of Reset pulse must be at least 50μs during or after power on. Subsequent Reset pulse can be 500 ns min.

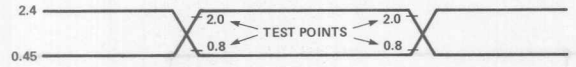


Figure 25. Input Waveforms for A.C. Tests

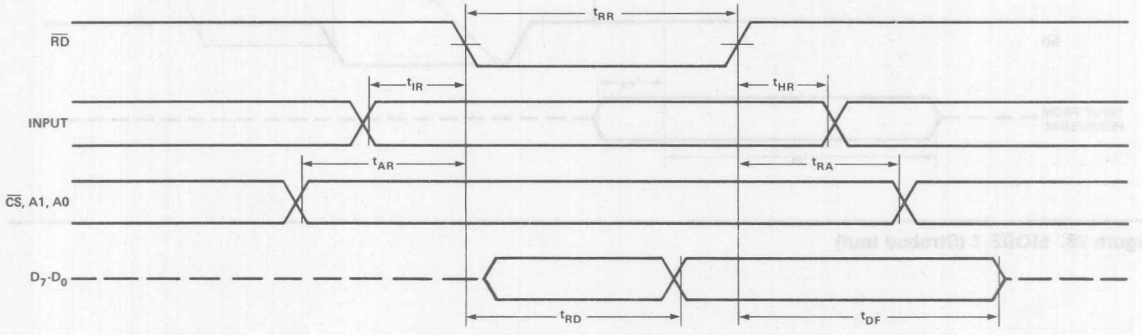


Figure 26. MODE 0 (Basic Input)

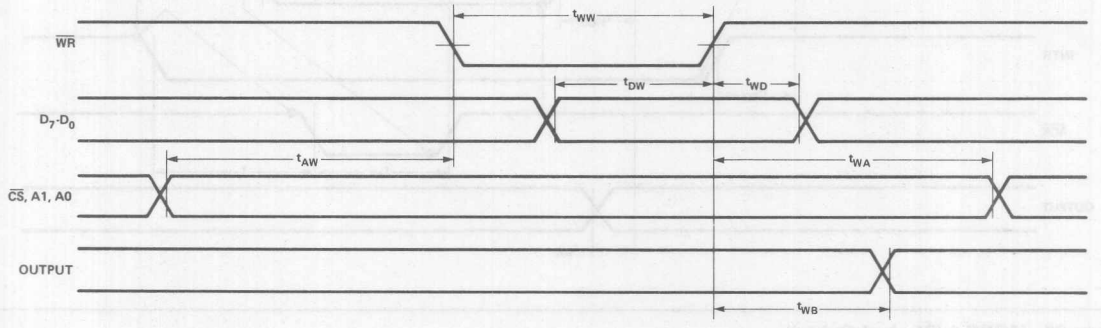


Figure 27. MODE 0 (Basic Output)

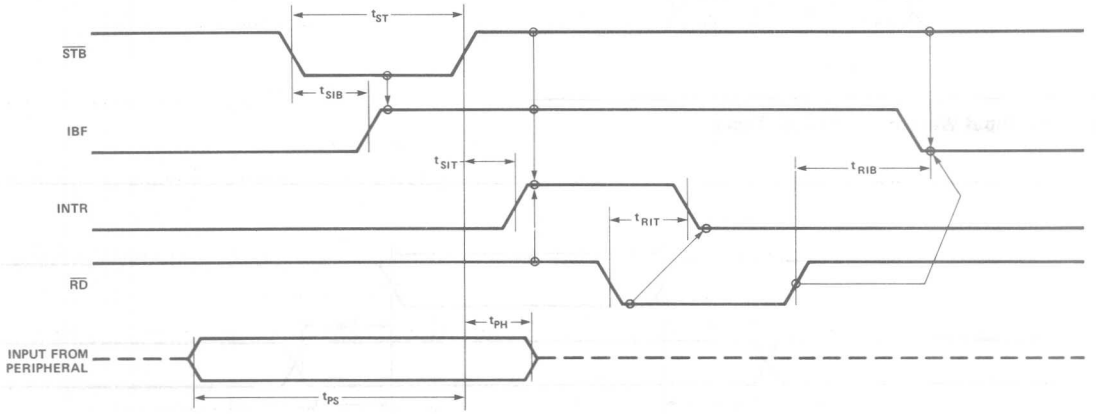


Figure 28. MODE 1 (Strobed Inut)

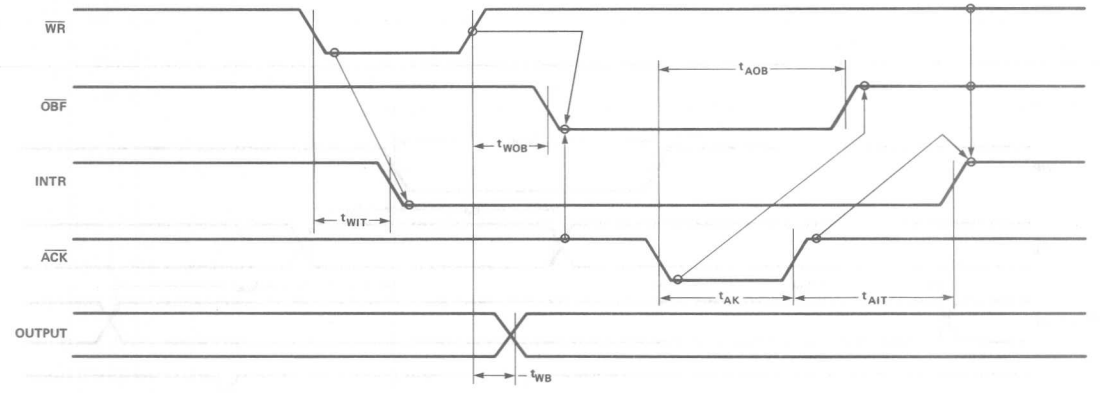


Figure 29. MODE 1 (Strobed Output)

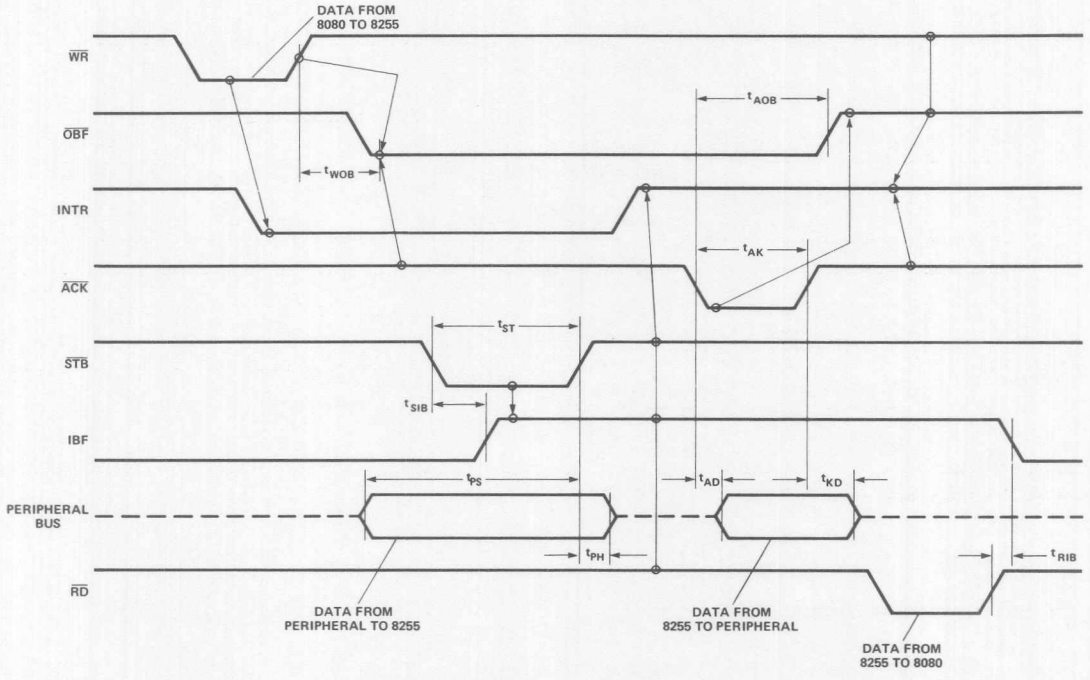


Figure 30. MODE 2 (Bidirectional)

NOTE: Any sequence where $\overline{WR}$ occurs before $\overline{ACK}$ and $\overline{STB}$ occurs before $\overline{RD}$ is permissible.
(INTR = IBF • $\overline{MASK}$ • $\overline{STB}$ • $\overline{RD}$ + $\overline{OBF}$ • $\overline{MASK}$ • $\overline{ACK}$ • $\overline{WR}$)

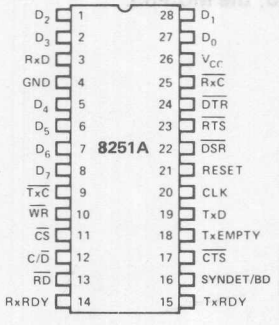
8251A
PROGRAMMABLE COMMUNICATION INTERFACE

PRELIMINARY
Notice: This is a product specification. Some parameters may be subject to change.

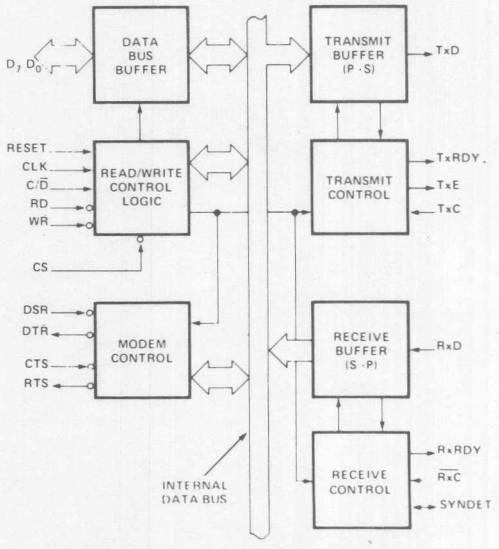
- **Synchronous and Asynchronous Operation**
- **Synchronous 5-8 Bit Characters; Internal or External Character Synchronization; Automatic Sync Insertion**
- **Asynchronous 5-8 Bit Characters; Clock Rate—1, 16 or 64 Times Baud Rate; Break Character Generation; 1, 1½, or 2 Stop Bits; False Start Bit Detection; Automatic Break Detect and Handling; 19.2K Baud.**
- **Baud Rate — DC to 64K Baud**
- **Full Duplex, Double Buffered, Transmitter and Receiver**
- **Error Detection — Parity, Overrun and Framing**
- **Fully Compatible with 8080/8085 CPU**
- **28-Pin DIP Package**
- **All Inputs and Outputs are TTL Compatible**
- **Single +5V Supply**
- **Single TTL Clock**

The Intel® 8251A is the enhanced version of the industry standard, Intel® 8251 Universal Synchronous/Asynchronous Receiver/Transmitter (USART), designed for data communications with Intel's new high performance family of microprocessors such as the 8085. The 8251A is used as a peripheral device and is programmed by the CPU to operate using virtually any serial data transmission technique presently in use (including IBM "bi-sync"). The USART accepts data characters from the CPU in parallel format and then converts them into a continuous serial data stream for transmission. Simultaneously, it can receive serial data streams and convert them into parallel data characters for the CPU. The USART will signal the CPU whenever it can accept a new character for transmission or whenever it has received a character for the CPU. The CPU can read the complete status of the USART at any time. These include data transmission errors and control signals such as SYNDET, TxEMPTY. The chip is constructed using N-channel silicon gate technology.

PIN CONFIGURATION



BLOCK DIAGRAM



PIN NAMES

D_7-D_0	Data Bus (8 bits)
C/D	Control or Data is to be Written or Read
RD	Read Data Command
WR	Write Data or Control Command
CS	Chip Enable
CLK	Clock Pulse (TTL)
RESET	Reset
TxC	Transmitter Clock
TxD	Transmitter Data
RxC	Receiver Clock
RxD	Receiver Data
RxRDY	Receiver Ready (has character for 8080)
TxRDY	Transmitter Ready (ready for char. from 8080)

DSR	Data Set Ready
DTR	Data Terminal Ready
SYNDET/BD	Sync Detect/ Break Detect
RTS	Request to Send Data
CTS	Clear to Send Data
TxE	Transmitter Empty
V_{CC}	+5 Volt Supply
GND	Ground

© Intel Corporation, 1978

FEATURES AND ENHANCEMENTS

8251A is an advanced design of the industry standard USART, the Intel® 8251. The 8251A operates with an extended range of Intel microprocessors that includes the new 8085 CPU and maintains compatibility with the 8251. Familiarization time is minimal because of compatibility and involves only knowing the additional features and enhancements, and reviewing the AC and DC specifications of the 8251A.

The 8251A incorporates all the key features of the 8251 and has the following additional features and enhancements:

- 8251A has double-buffered data paths with separate I/O registers for control, status, Data In, and Data Out, which considerably simplifies control programming and minimizes CPU overhead.
- In asynchronous operations, the Receiver detects and handles "break" automatically, relieving the CPU of this task.
- A refined Rx initialization prevents the Receiver from starting when in "break" state, preventing unwanted interrupts from a disconnected USART.
- At the conclusion of a transmission, TxD line will always return to the marking state unless SBRK is programmed.
- Tx Enable logic enhancement prevents a Tx Disable command from halting transmission until all data previously written has been transmitted. The logic also prevents the transmitter from turning off in the middle of a word.
- When External Sync Detect is programmed, Internal Sync Detect is disabled, and an External Sync Detect status is provided via a flip-flop which clears itself upon a status read.
- Possibility of false sync detect is minimized by ensuring that if double character sync is programmed, the characters be contiguously detected and also by clearing the Rx register to all ones whenever Enter Hunt command is issued in Sync mode.
- As long as the 8251A is not selected, the $\overline{RD}$ and $\overline{WR}$ do not affect the internal operation of the device.
- The 8251A Status can be read at any time but the status update will be inhibited during status read.
- The 8251A is free from extraneous glitches and has enhanced AC and DC characteristics, providing higher speed and better operating margins.
- Baud rate from DC to 64K.
- Fully compatible with Intel's new industry standard, the MCS-85.

8251A BASIC FUNCTIONAL DESCRIPTION

General

The 8251A is a Universal Synchronous/Asynchronous Receiver/Transmitter designed specifically for the 80/85 Microcomputer Systems. Like other I/O devices in a Microcomputer System, its functional configuration is programmed by the system's software for maximum flexibility. The 8251A can support virtually any serial data technique currently in use including bi-sync.

In a communication environment an interface device must convert parallel format system data into serial format for transmission and convert incoming serial format data into parallel system data for reception. The interface device must also delete or insert bits or characters that are functionally unique to the communication technique. In essence, the interface should appear "transparent" to the CPU, a simple input or output of byte-oriented system data.

Data Bus Buffer

This 3-state, bidirectional, 8-bit buffer is used to interface the 8251A to the system Data Bus. Data is transmitted or received by the buffer upon execution of INput or OUTput instructions of the CPU. Control words, Command words and Status information are also transferred through the Data Bus Buffer. The command status and data in, and data out are separate 8-bit registers to provide double buffering.

This functional block accepts inputs from the system Control bus and generates control signals for overall device operation. It contains the Control Word Register and Command Word Register that store the various control formats for the device functional definition.

RESET (Reset)

A "high" on this input forces the 8251A into an "Idle" mode. The device will remain at "Idle" until a new set of control words is written into the 8251A to program its functional definition. Minimum RESET pulse width is 6 t_{CY} (clock must be running).

CLK (Clock)

The CLK input is used to generate internal device timing and is normally connected to the Phase 2 (TTL) output of the 8224 Clock Generator. No external inputs or outputs are referenced to CLK but the frequency of CLK must be greater than 30 times the Receiver or Transmitter data bit rates.

$\overline{WR}$ (Write)

A "low" on this input informs the 8251A that the CPU is writing data or control words to the 8251A.

$\overline{RD}$ (Read)

A "low" on this input informs the 8251A that the CPU is reading data or status information from the 8251A.

C/$\overline{D}$ (Control/Data)

This input, in conjunction with the $\overline{WR}$ and $\overline{RD}$ inputs, informs the 8251A that the word on the Data Bus is either a data character, control word or status information.
1 = CONTROL/STATUS 0 = DATA

$\overline{CS}$ (Chip Select)

A "low" on this input selects the 8251A. No reading or writing will occur unless the device is selected. When $\overline{CS}$ is high, the Data Bus in the float state and $\overline{RD}$ and $\overline{WR}$ will have no effect on the chip.

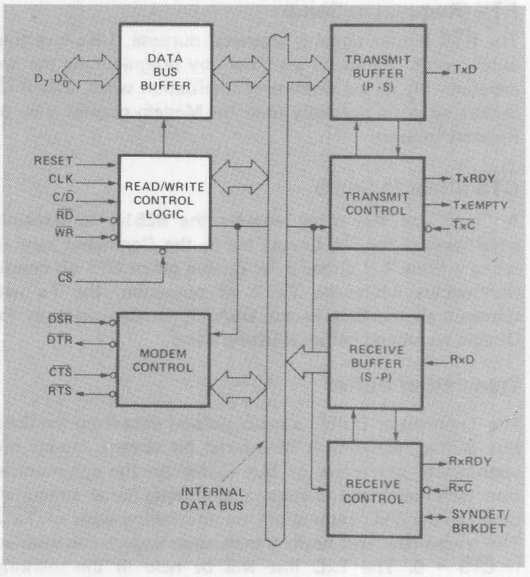


Figure 1. 8251A Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

C/$\overline{D}$	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	
0	0	1	0	8251A DATA ⇒ DATA BUS
0	1	0	0	DATA BUS ⇒ 8251A DATA
1	0	1	0	STATUS ⇒ DATA BUS
1	1	0	0	DATA BUS ⇒ CONTROL
X	1	1	0	DATA BUS ⇒ 3-STATE
X	X	X	1	DATA BUS ⇒ 3-STATE

Modem Control

The 8251A has a set of control inputs and outputs that can be used to simplify the interface to almost any modem. The modem control signals are general purpose in nature and can be used for functions other than modem control, if necessary.

$\overline{DSR}$ (Data Set Ready)

The $\overline{DSR}$ input signal is a general purpose, 1-bit inverting input port. Its condition can be tested by the CPU using a Status Read operation. The $\overline{DSR}$ input is normally used to test modem conditions such as Data Set Ready.

$\overline{DTR}$ (Data Terminal Ready)

The $\overline{DTR}$ output signal is a general purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{DTR}$ output signal is normally used for modem control such as Data Terminal Ready or Rate Select.

$\overline{RTS}$ (Request to Send)

The $\overline{RTS}$ output signal is a general purpose, 1-bit inverting output port. It can be set "low" by programming the appropriate bit in the Command Instruction word. The $\overline{RTS}$ output signal is normally used for Modem control such as Request to Send.

$\overline{CTS}$ (Clear to Send)

A "low" on this input enables the 8251A to transmit serial data if the Tx Enable bit in the Command byte is set to a "one." If either a Tx Enable off or CTS off condition occurs while the Tx is in operation, the Tx will transmit all the data in the USART, written prior to Tx Disable command before shutting down.

Transmitter Buffer

The Transmitter Buffer accepts parallel data from the Data Bus Buffer, converts it to a serial bit stream, inserts the appropriate characters or bits (based on the communication technique) and outputs a composite serial stream of data on the TxD output pin on the falling edge of $\overline{TxC}$. The transmitter will begin transmission upon being enabled if $\overline{CTS}$ = 0. The TxD line will be held in the marking state immediately upon a master Reset or when Tx Enable/$\overline{CTS}$ off or TxEMPTY.

Transmitter Control

The transmitter Control manages all activities associated with the transmission of serial data. It accepts and issues signals both externally and internally to accomplish this function.

TxRDY (Transmitter Ready)

This output signals the CPU that the transmitter is ready to accept a data character. The TxRDY output pin can be used as an interrupt to the system, since it is masked by Tx Disabled, or, for Polled operation, the CPU can check TxRDY using a Status Read operation. TxRDY is automatically reset by the leading edge of $\overline{WR}$ when a data character is loaded from the CPU.

Note that when using the Polled operation, the TxRDY status bit is *not* masked by Tx Enabled, but will only indicate the Empty/Full Status of the Tx Data Input Register.

TxE (Transmitter Empty)

When the 8251A has no characters to transmit, the TxEMPTY output will go "high". It resets automatically upon receiving a character from the CPU. TxEMPTY can be used to indicate the end of a transmission mode, so that the CPU "knows" when to "turn the line around" in the half-duplexed operational mode. TxEMPTY is independent of the Tx Enable bit in the Command instruction.

In SYNChronous mode, a "high" on this output indicates that a character has not been loaded and the SYNC character or characters are about to be or are being transmitted automatically as "fillers". TxEMPTY does not go low when the SYNC characters are being shifted out.

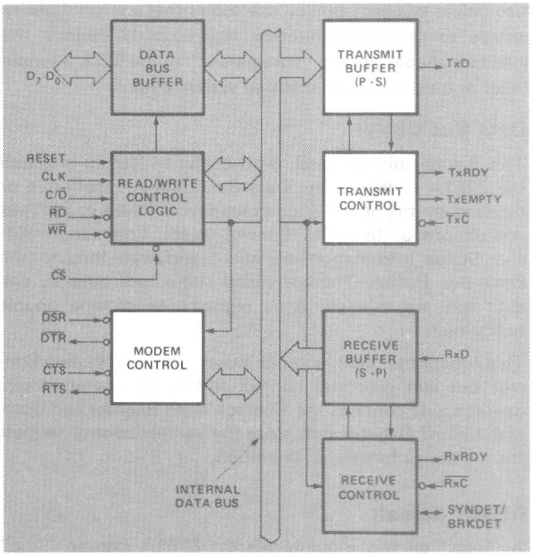


Figure 2. 8251A Block Diagram Showing Modem and Transmitter Buffer and Control Functions

$\overline{TxC}$ (Transmitter Clock)

The Transmitter Clock controls the rate at which the character is to be transmitted. In the Synchronous transmission mode, the Baud Rate (1x) is equal to the $\overline{TxC}$ frequency. In Asynchronous transmission mode the baud rate is a fraction of the actual $\overline{TxC}$ frequency. A portion of the mode instruction selects this factor; it can be 1, 1/16 or 1/64 the $\overline{TxC}$.

For Example:

If Baud Rate equals 110 Baud,
$\overline{TxC}$ equals 110 Hz (1x)
$\overline{TxC}$ equals 1.76 kHz (16x)
$\overline{TxC}$ equals 7.04 kHz (64x).

The falling edge of $\overline{TxC}$ shifts the serial data out of the 8251A.

Receiver Buffer

The Receiver accepts serial data, converts this serial input to parallel format, checks for bits or characters that are unique to the communication technique and sends an "assembled" character to the CPU. Serial data is input to RxD pin, and is clocked in on the rising edge of $\overline{RxC}$.

Receiver Control

This functional block manages all receiver-related activities which consist of the following features:

The RxD initialization circuit prevents the 8251A from mistaking an unused input line for an active low data line in the "break condition". Before starting to receive serial characters on the RxD line, a valid "1" must first be detected after a chip master Reset. Once this has been determined, a search for a valid low (Start bit) is enabled. This feature is only active in the asynchronous mode, and is only done once for each master Reset.

The False Start bit detection circuit prevents false starts due to a transient noise spike by first detecting the falling edge and then strobing the nominal center of the Start bit (RxD = low).

The Parity Toggle F/F and Parity Error F/F circuits are used for parity error detection and set the corresponding status bit.

The Framing Error Flag F/F is set if the Stop bit is absent at the end of the data byte (asynchronous mode), and also sets the corresponding status bit.

RxRDY (Receiver Ready)

This output indicates that the 8251A contains a character that is ready to be input to the CPU. Rx RDY can be connected to the interrupt structure of the CPU or, for Polled operation, the CPU can check the condition of RxRDY using a Status Read operation.

Rx Enable off both masks and holds RxRDY in the Reset Condition. For Asynchronous mode, to set RxRDY, the Receiver must be Enabled to sense a Start Bit and a complete character must be assembled and transferred to the Data Output Register. For Synchronous mode, to set RxRDY, the Receiver must be enabled and a character must finish assembly and be transferred to the Data Output Register.

Failure to read the received character from the Rx Data Output Register prior to the assembly of the next Rx Data character will set overrun condition error and the previous character will be written over and lost. If the Rx Data is being read by the CPU when the internal transfer is occurring, overrun error will be set and the old character will be lost.

$\overline{RxC}$ (Receiver Clock)

The Receiver Clock controls the rate at which the character is to be received. In Synchronous Mode, the Baud Rate (1x) is equal to the actual frequency of $\overline{RxC}$. In Asynchronous Mode, the Baud Rate is a fraction of the actual $\overline{RxC}$ frequency. A portion of the mode instruction selects this factor; 1, 1/16 or 1/64 the $\overline{RxC}$.

For Example:

Baud Rate equals 300 Baud, if
$\overline{RxC}$ equals 300 Hz (1x)
$\overline{RxC}$ equals 4800 Hz (16x)
$\overline{RxC}$ equals 19.2 kHz (64x).

Baud Rate equals 2400 Baud, if
$\overline{RxC}$ equals 2400 Hz (1x)
$\overline{RxC}$ equals 38.4 kHz (16x)
$\overline{RxC}$ equals 153.6 kHz (64x).

Data is sampled into the 8251A on the rising edge of $\overline{RxC}$.

NOTE: In most communications systems, the 8251A will be handling both the transmission and reception operations of a single link. Consequently, the Receive and Transmit Baud Rates will be the same. Both $\overline{TxC}$ and $\overline{RxC}$ will require identical frequencies for this operation and can be tied together and connected to a single frequency source (Baud Rate Generator) to simplify the interface.

SYNDET (SYNC Detect)/BRKDET (Break Detect)

This pin is used in SYNChronous Mode for SYNDET and may be used as either input or output, programmable through the Control Word. It is reset to output mode low upon RESET. When used as an output (internal Sync mode), the SYNDET pin will go "high" to indicate that the 8251A has located the SYNC character in the Receive mode. If the 8251A is programmed to use double Sync characters (bisync), then SYNDET will go "high" in the middle of the last bit of the second Sync character. SYNDET is automatically reset upon a Status Read operation.

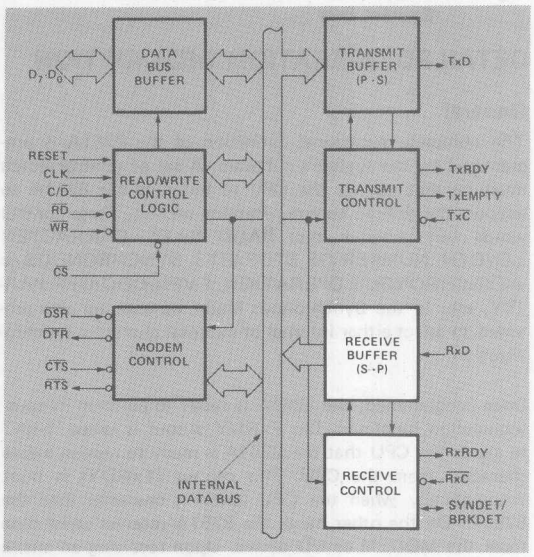


Figure 3. 8251A Block Diagram Showing Receiver Buffer and Control Functions

When used as an input (external SYNC detect mode), a positive going signal will cause the 8251A to start assembling data characters on the rising edge of the next $\overline{RxC}$. Once in SYNC, the "high" input signal can be removed. the period of $\overline{RxC}$. When External SYNC Detect is programmed, the Internal SYNC Detect is disabled.

BREAK DETECT (Async Mode Only)

This output will go high whenever an all zero word of the programmed length (including start bit, data bit, parity bit, and *one* stop bit) is received. Break Detect may also be read as a Status bit. It is reset only upon a master chip Reset or Rx Data returning to a "one" state.

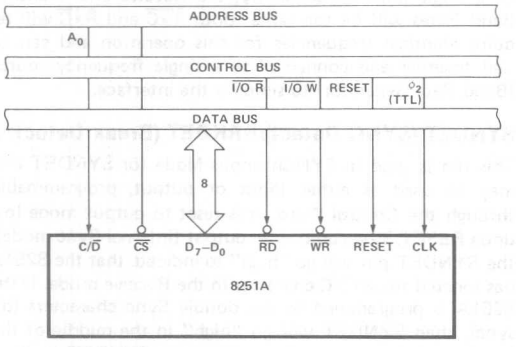


Figure 4. 8251A Interface to 8080 Standard System Bus

DETAILED OPERATION DESCRIPTION

General

The complete functional definition of the 8251A is programmed by the system's software. A set of control words must be sent out by the CPU to initialize the 8251A to support the desired communications format. These control words will program the: BAUD RATE, CHARACTER LENGTH, NUMBER OF STOP BITS, SYNCHRONOUS or ASYNCHRONOUS OPERATION, EVEN/ODD/OFF PARITY, etc. In the Synchronous Mode, options are also provided to select either internal or external character synchronization.

Once programmed, the 8251A is ready to perform its communication functions. The TxRDY output is raised "high" to signal the CPU that the 8251A is ready to receive a data character from the CPU. This output (TxRDY) is reset automatically when the CPU writes a character into the 8251A. On the other hand, the 8251A receives serial data from the MODEM or I/O device. Upon receiving an entire character, the RxRDY output is raised "high" to signal the CPU that the 8251A has a complete character ready for the CPU to fetch. RxRDY is reset automatically upon the CPU data read operation.

The 8251A cannot begin transmission until the Tx Enable (Transmitter Enable) bit is set in the Command Instruction and it has received a Clear To Send ($\overline{CTS}$) input. The TxD output will be held in the marking state upon Reset.

Programming the 8251A

Prior to starting data transmission or reception, the 8251A must be loaded with a set of control words generated by the CPU. These control signals define the complete functional definition of the 8251A and must immediately follow a Reset operation (internal or external).

The control words are split into two formats:

1. Mode Instruction
2. Command Instruction

Mode Instruction

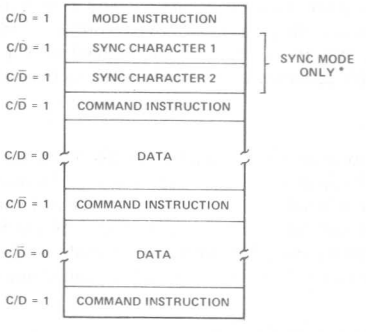
This format defines the general operational characteristics of the 8251A. It must follow a Reset operation (internal or external). Once the Mode Instruction has been written into the 8251A by the CPU, SYNC characters or Command Instructions may be inserted.

Command Instruction

This format defines a status word that is used to control the actual operation of the 8251A.

Both the Mode and Command Instructions must conform to a specified sequence for proper device operation. The Mode Instruction must be inserted immediately following a Reset operation, prior to using the 8251A for data communication.

All control words written into the 8251A after the Mode Instruction will load the Command Instruction. Command Instructions can be written into the 8251A at any time in the data block during the operation of the 8251A. To return to the Mode Instruction format, the master Reset bit in the Command Instruction word can be set to initiate an internal Reset operation which automatically places the 8251A back into the Mode Instruction format. Command Instructions must follow the Mode Instructions or Sync characters.



* The second SYNC character is skipped if MODE instruction has programmed the 8251A to single character Internal SYNC Mode. Both SYNC characters are skipped if MODE instruction has programmed the 8251A to ASYNC mode.

Figure 5. Typical Data Block

Mode Instruction Definition

The 8251A can be used for either Asynchronous or Synchronous data communication. To understand how the Mode Instruction defines the functional operation of the 8251A, the designer can best view the device as two separate components sharing the same package, one Asynchronous the other Synchronous. The format definition can be changed only after a master chip Reset. For explanation purposes the two formats will be isolated.

NOTE: When parity is enabled it is not considered as one of the data bits for the purpose of programming the word length. The actual parity bit received on the Rx Data line cannot be read on the Data Bus. In the case of a programmed character length of less than 8 bits, the least significant Data Bus bits will hold the data; unused bits are "don't care" when writing data to the 8251A, and will be "zeros" when reading the data from the 8251A.

Asynchronous Mode (Transmission)

Whenever a data character is sent by the CPU the 8251A automatically adds a Start bit (low level) followed by the data bits (least significant bit first), and the programmed number of Stop bits to each character. Also, an even or odd Parity bit is inserted prior to the Stop bit(s), as defined by the Mode Instruction. The character is then transmitted as a serial data stream on the TxD output. The serial data is shifted out on the falling edge of $\overline{TxC}$ at a rate equal to 1, 1/16, or 1/64 that of the $\overline{TxC}$, as defined by the Mode Instruction. BREAK characters can be continuously sent to the TxD if commanded to do so.

When no data characters have been loaded into the 8251A the TxD output remains "high" (marking) unless a Break (continuously low) has been programmed.

Asynchronous Mode (Receive)

The RxD line is normally high. A falling edge on this line triggers the beginning of a START bit. The validity of this START bit is checked by again strobing this bit at its nominal center (16X or 64X mode only). If a low is detected again, it is a valid START bit, and the bit counter will start counting. The bit counter thus locates the center of the data bits, the parity bit (if it exists) and the stop bits. If parity error occurs, the parity error flag is set. Data and parity bits are sampled on the RxD pin with the rising edge of $\overline{RxC}$. If a low level is detected as the STOP bit, the Framing Error flag will be set. The STOP bit signals the end of a character. Note that the *receiver* requires only *one* stop bit, regardless of the number of stop bits programmed. This character is then loaded into the parallel I/O buffer of the 8251A. The RxRDY pin is raised to signal the CPU that a character is ready to be fetched. If a previous character has not been fetched by the CPU, the present character replaces it in the I/O buffer, and the OVERRUN Error flag is raised (thus the previous character is lost). All of the error flags can be reset by an Error Reset Instruction. The occurrence of any of these errors will not affect the operation of the 8251A.

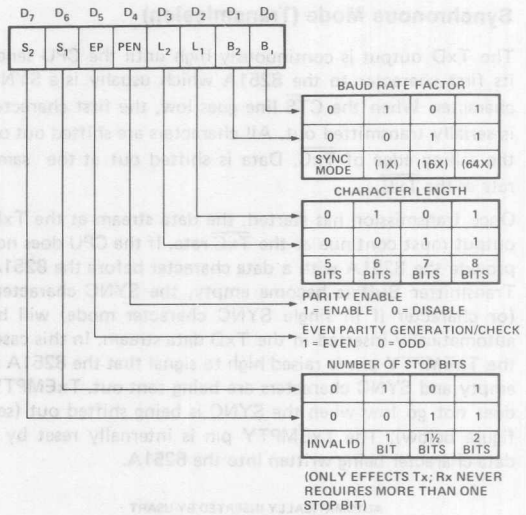


Figure 6. Mode Instruction Format, Asynchronous Mode

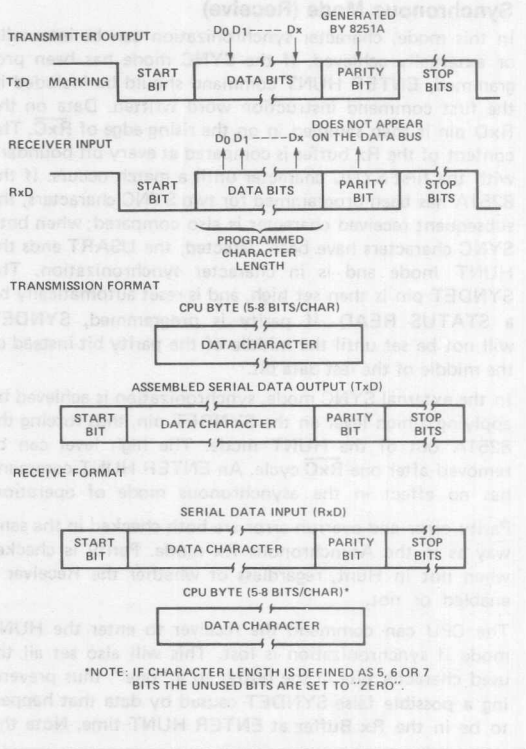
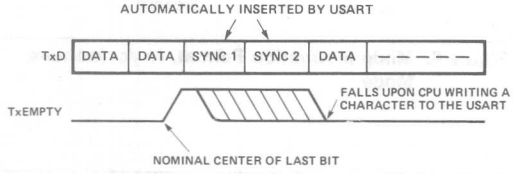


Figure 7. Asynchronous Mode

Synchronous Mode (Transmission)

The TxD output is continuously high until the CPU sends its first character to the 8251A which usually is a SYNC character. When the $\overline{CTS}$ line goes low, the first character is serially transmitted out. All characters are shifted out on the falling edge of $\overline{TxC}$. Data is shifted out at the same rate as the $\overline{TxC}$.

Once transmission has started, the data stream at the TxD output must continue at the $\overline{TxC}$ rate. If the CPU does not provide the 8251A with a data character before the 8251A Transmitter Buffers become empty, the SYNC characters (or character if in single SYNC character mode) will be automatically inserted in the TxD data stream. In this case, the TxEMPTY pin is raised high to signal that the 8251A is empty and SYNC characters are being sent out. TxEMPTY does not go low when the SYNC is being shifted out (see figure below). The TxEMPTY pin is internally reset by a data character being written into the 8251A.



Synchronous Mode (Receive)

In this mode, character synchronization can be internally or externally achieved. If the SYNC mode has been programmed, ENTER HUNT command should be included in the first command instruction word written. Data on the RxD pin is then sampled in on the rising edge of $\overline{RxC}$. The content of the Rx buffer is compared at every bit boundary with the first SYNC character until a match occurs. If the 8251A has been programmed for two SYNC characters, the subsequent received character is also compared; when both SYNC characters have been detected, the USART ends the HUNT mode and is in character synchronization. The SYNDET pin is then set high, and is reset automatically by a STATUS READ. If parity is programmed, SYNDET will not be set until the middle of the parity bit instead of the middle of the last data bit.

In the external SYNC mode, synchronization is achieved by applying a high level on the SYNDET pin, thus forcing the 8251A out of the HUNT mode. The high level can be removed after one $\overline{RxC}$ cycle. An ENTER HUNT command has no effect in the asynchronous mode of operation.

Parity error and overrun error are both checked in the same way as in the Asynchronous Rx mode. Parity is checked when not in Hunt, regardless of whether the Receiver is enabled or not.

The CPU can command the receiver to enter the HUNT mode if synchronization is lost. This will also set all the used character bits in the buffer to a "one", thus preventing a possible false SYNDET caused by data that happens to be in the Rx Buffer at ENTER HUNT time. Note that the SYNDET F/F is reset at each Status Read, regardless of whether internal or external SYNC has been programmed. This does not cause the 8251A to return to the HUNT mode. When in SYNC mode, but not in HUNT, Sync Detection is still functional, but only occurs at the "known" word boundaries. Thus, if one Status Read indicates SYNDET and a second Status Read also indicates SYNDET, then the programmed SYNDET characters have been received since the previous Status Read. (If double character sync has been programmed, then both sync characters have been contiguously received to gate a SYNDET indication.) When external SYNDET mode is selected, internal Sync Detect is disabled, and the SYNDET F/F may be set at any bit boundary.

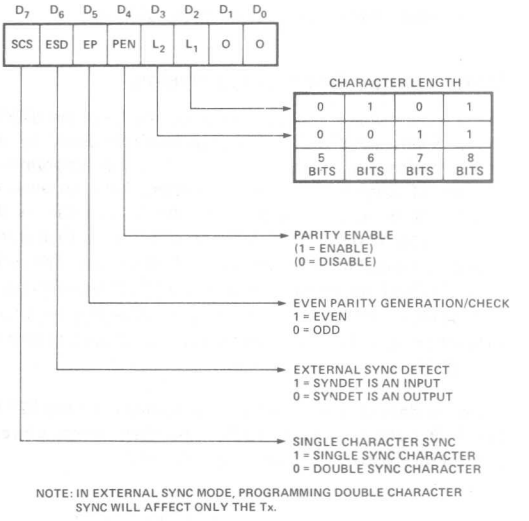


Figure 8. Mode Instruction Format

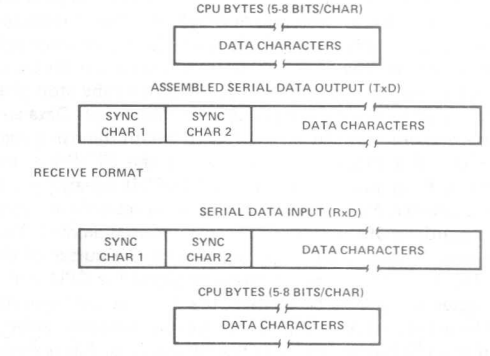
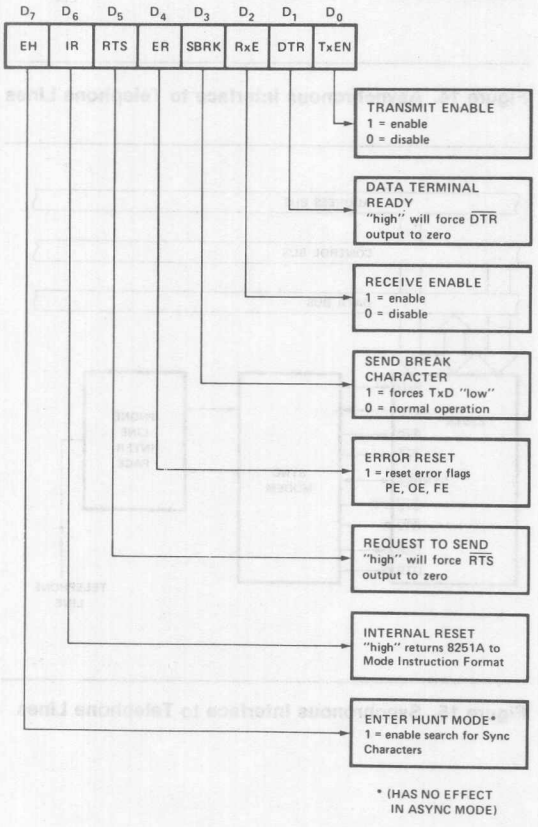


Figure 9. Data Format, Synchronous Mode

COMMAND INSTRUCTION DEFINITION

Once the functional definition of the 8251A has been programmed by the Mode Instruction and the Sync Characters are loaded (if in Sync Mode) then the device is ready to be used for data communication. The Command Instruction controls the actual operation of the selected format. Functions such as: Enable Transmit/Receive, Error Reset and Modem Controls are provided by the Command Instruction.

Once the Mode Instruction has been written into the 8251A and Sync characters inserted, if necessary, then all further "control writes" ($C/\overline{D}$ = 1) will load a Command Instruction. A Reset Operation (internal or external) will return the 8251A to the Mode Instruction format.



Note: Error Reset must be performed whenever RxEnable and Enter Hunt are programmed.

Figure 10. Command Instruction Format

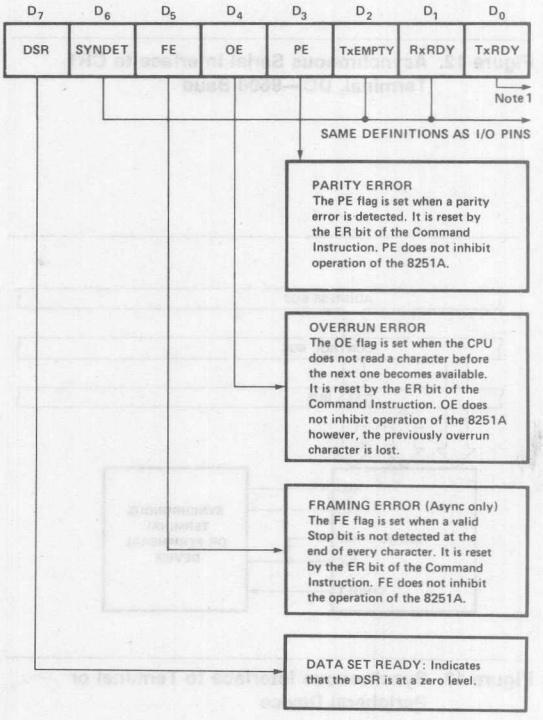
STATUS READ DEFINITION

In data communication systems it is often necessary to examine the "status" of the active device to ascertain if errors have occurred or other conditions that require the processor's attention. The 8251A has facilities that allow the programmer to "read" the status of the device at any time during the functional operation. (The status update is inhibited during status read).

A normal "read" command is issued by the CPU with $C/\overline{D}$ = 1 to accomplish this function.

Some of the bits in the Status Read Format have identical meanings to external output pins so that the 8251A can be used in a completely Polled environment or in an interrupt driven environment. TxRDY is an exception.

Note that status update can have a maximum delay of 28 clock periods from the actual event affecting the status.



Note 1: TxRDY status bit has different meanings from the TxRDY output pin. The former is not conditioned by $\overline{CTS}$ and TxEN; the latter is conditioned by both $\overline{CTS}$ and TxEN.

i.e. TxRDY status bit = DB Buffer Empty

TxRDY pin out = DB Buffer Empty · (CTS=0) · (TxEN=1)

Figure 11. Status Read Format

APPLICATIONS OF THE 8251A

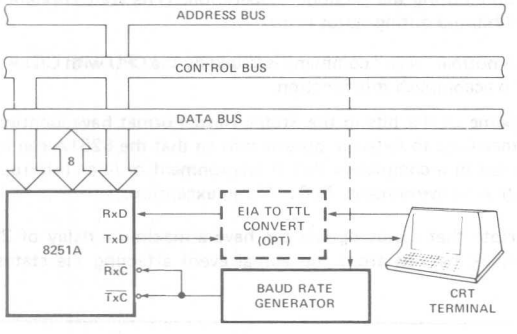


Figure 12. Asynchronous Serial Interface to CRT Terminal, DC—9600 Baud

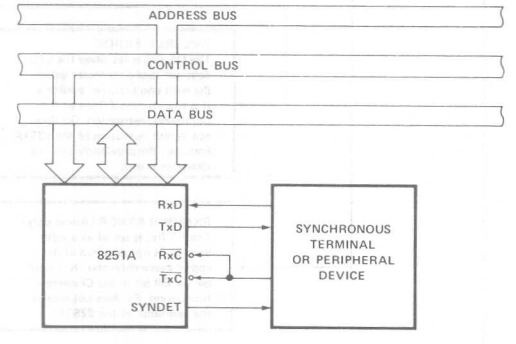


Figure 13. Synchronous Interface to Terminal or Peripheral Device

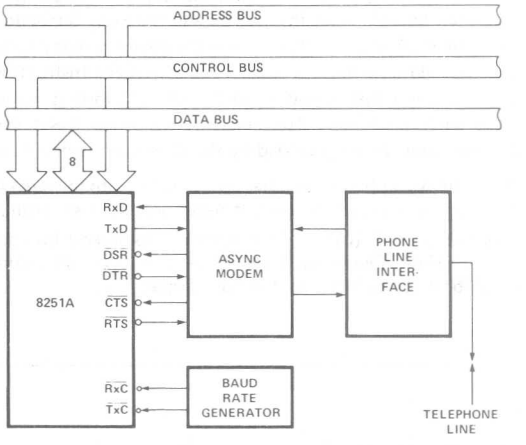


Figure 14. Asynchronous Interface to Telephone Lines

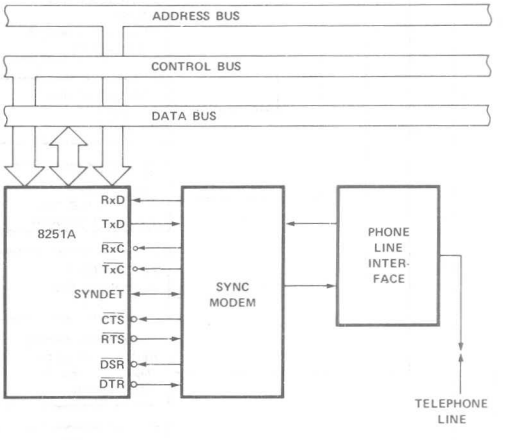


Figure 15. Synchronous Interface to Telephone Lines

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias. 0°C to 70°C
Storage Temperature –65°C to +150°C
Voltage On Any Pin
With Respect to Ground –0.5V to +7V
Power Dissipation 1 Watt

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS

T_A = 0°C to 70°C; V_{CC} = 5.0V ±5%; GND = 0V

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL}	Input Low Voltage	–0.5	0.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}	V	
V_{OL}	Output Low Voltage		0.45	V	I_{OL} = 2.2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = –400 μA
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} TO 0.45V
I_{IL}	Input Leakage		±10	μA	V_{IN} = V_{CC} TO 0.45V
I_{CC}	Power Supply Current		100	mA	All Outputs = High

CAPACITANCE

T_A = 25°C; V_{CC} = GND = 0V

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance		10	pF	fc = 1MHz
$C_{I/O}$	I/O Capacitance		20	pF	Unmeasured pins returned to GND

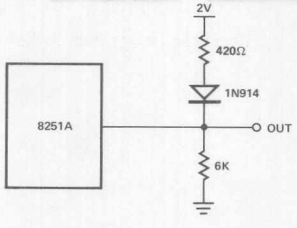


Figure 16. Test Load Circuit

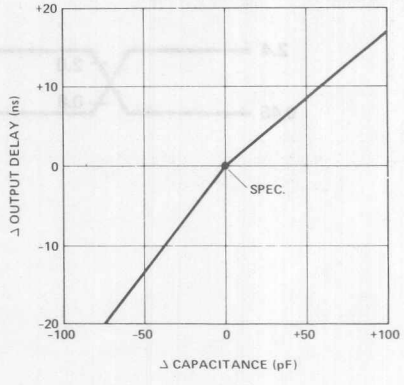


Figure 17. Typical Δ Output Delay vs. Δ Capacitance (pF)

A.C. CHARACTERISTICS

T_A = 0°C to 70°C; V_{CC} = 5.0V ±5%; GND = 0V

Bus Parameters (Note 1)

Read Cycle:

SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITIONS
t_{AR}	Address Stable Before $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	0		ns	Note 2
t_{RA}	Address Hold Time for $\overline{READ}$ ($\overline{CS}$, C/$\overline{D}$)	0		ns	Note 2
t_{RR}	$\overline{READ}$ Pulse Width	250		ns	
t_{RD}	Data Delay from $\overline{READ}$		200	ns	3, C_L = 150 pF
t_{DF}	$\overline{READ}$ to Data Floating	10	100	ns	

Write Cycle:

SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITIONS
t_{AW}	Address Stable Before $\overline{WRITE}$	0		ns	
t_{WA}	Address Hold Time for $\overline{WRITE}$	0		ns	
t_{WW}	$\overline{WRITE}$ Pulse Width	250		ns	
t_{DW}	Data Set Up Time for $\overline{WRITE}$	150		ns	
t_{WD}	Data Hold Time for $\overline{WRITE}$	0		ns	
t_{RV}	Recovery Time Between WRITES	6		t_{CY}	Note 4

NOTES: 1. AC timings measured V_{OH} = 2.0, V_{OL} = 0.8, and with load circuit of Figure 1.
2. Chip Select ($\overline{CS}$) and Command/Data (C/$\overline{D}$) are considered as Addresses.
3. Assumes that Address is valid before $\overline{RD}\downarrow$.
4. This recovery time is for Mode Initialization only. Write Data is allowed only when TxRDY = 1.
Recovery Time between Writes for Asynchronous Mode is 8 t_{CY} and for Synchronous Mode is 16 t_{CY}.

Input Waveforms for AC Tests



Other Timings:

SYMBOL	PARAMETER	MIN.	MAX.	UNIT	TEST CONDITIONS
t_{CY}	Clock Period	320	1.35	μs	Notes 5, 6
t_{ϕ}	Clock High Pulse Width	120	t_{CY-90}	ns	
$t_{\overline{\phi}}$	Clock Low Pulse Width	90		ns	
t_R, t_F	Clock Rise and Fall Time	5	20	ns	
t_{DTx}	TxD Delay from Falling Edge of $\overline{TxC}$		1	μs	
t_{SRx}	Rx Data Set-Up Time to Sampling Pulse	2		μs	
t_{HRx}	Rx Data Hold Time to Sampling Pulse	2		μs	
f_{Tx}	Transmitter Input Clock Frequency				
	1x Baud Rate	DC	64	kHz	
	16x Baud Rate	DC	310	kHz	
	64x Baud Rate	DC	615	kHz	
t_{TPW}	Transmitter Input Clock Pulse Width				
	1x Baud Rate	12		t_{CY}	
	16x and 64x Baud Rate	1		t_{CY}	
t_{TPD}	Transmitter Input Clock Pulse Delay				
	1x Baud Rate	15		t_{CY}	
	16x and 64x Baud Rate	3		t_{CY}	
f_{Rx}	Receiver Input Clock Frequency				
	1x Baud Rate	DC	64	kHz	
	16x Baud Rate	DC	310	kHz	
	64x Baud Rate	DC	615	kHz	
t_{RPW}	Receiver Input Clock Pulse Width				
	1x Baud Rate	12		t_{CY}	
	16x and 64x Baud Rate	1		t_{CY}	
t_{RPD}	Receiver Input Clock Pulse Delay				
	1x Baud Rate	15		t_{CY}	
	16x and 64x Baud Rate	3		t_{CY}	
t_{TxRDY}	TxRDY Pin Delay from Center of last Bit		8	t_{CY}	Note 7
$t_{TxRDY\ CLEAR}$	TxRDY ↓ from Leading Edge of $\overline{WR}$		150	ns	Note 7
t_{RxRDY}	RxRDY Pin Delay from Center of last Bit		24	t_{CY}	Note 7
$t_{RxRDY\ CLEAR}$	RxRDY ↓ from Leading Edge of $\overline{RD}$		150	ns	Note 7
t_{IS}	Internal SYNDET Delay from Rising Edge of $\overline{RxC}$		24	t_{CY}	Note 7
t_{ES}	External SYNDET Set-Up Time Before Falling Edge of $\overline{RxC}$		16	t_{CY}	Note 7
$t_{TxEMPTY}$	TxEMPTY Delay from Center of Data Bit		20	t_{CY}	Note 7
t_{WC}	Control Delay from Rising Edge of WRITE (TxEn, $\overline{DTR}$, $\overline{RTS}$)		8	t_{CY}	Note 7
t_{CR}	Control to READ Set-Up Time ($\overline{DSR}$, $\overline{CTS}$)		20	t_{CY}	Note 7

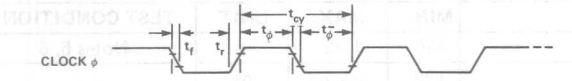
5. The TxC and RxC frequencies have the following limitations with respect to CLK.
 For 1x Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(30\ t_{CY})$
 For 16x and 64x Baud Rate, f_{Tx} or $f_{Rx} \leq 1/(4.5\ t_{CY})$

6. Reset Pulse Width = 6 t_{CY} minimum; System Clock must be running during Reset.

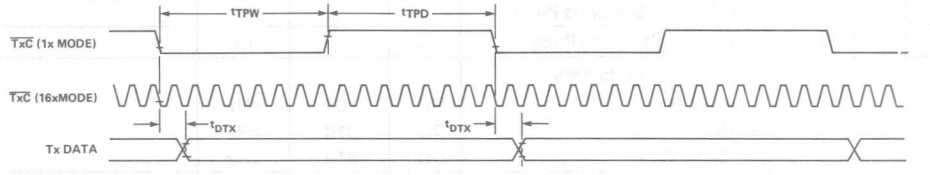
7. Status update can have a maximum delay of 28 clock periods from the event affecting the status.

WAVEFORMS

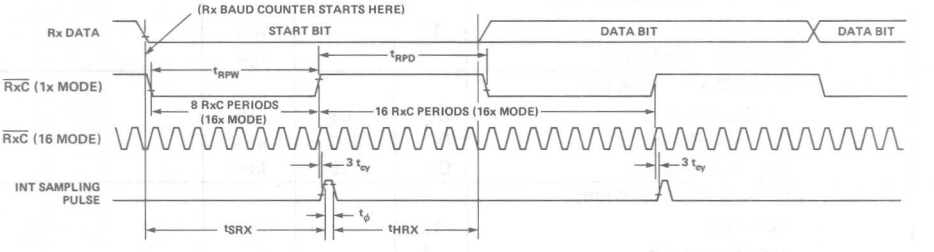
System Clock Input



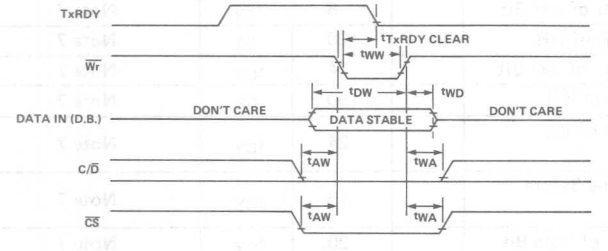
Transmitter Clock & Data



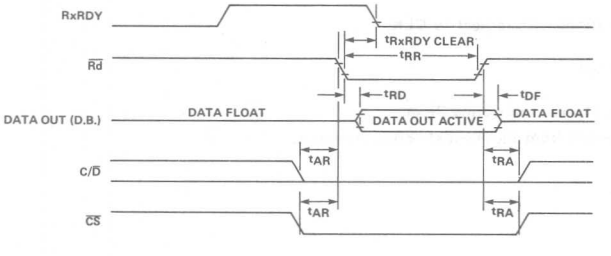
Receiver Clock & Data



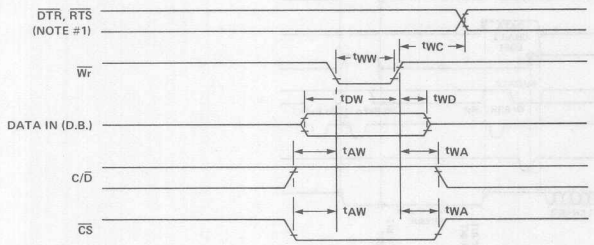
Write Data Cycle (CPU → USART)



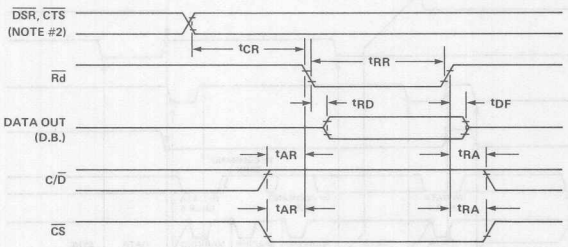
Read Data Cycle (CPU ← USART)



Write Control or Output Port Cycle (CPU → USART)

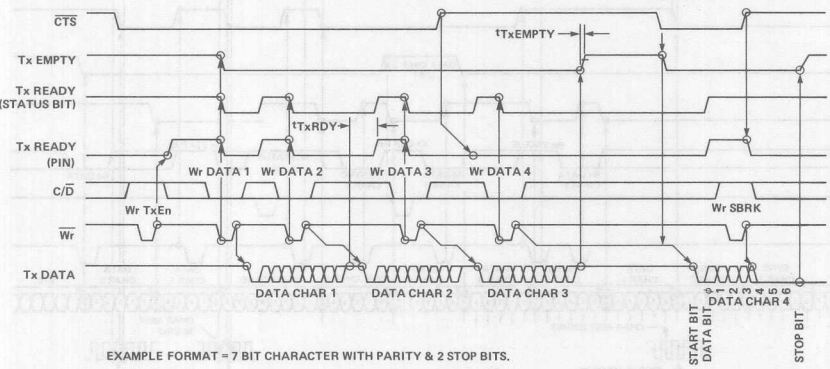


Read Control or Input Port (CPU ← USART)



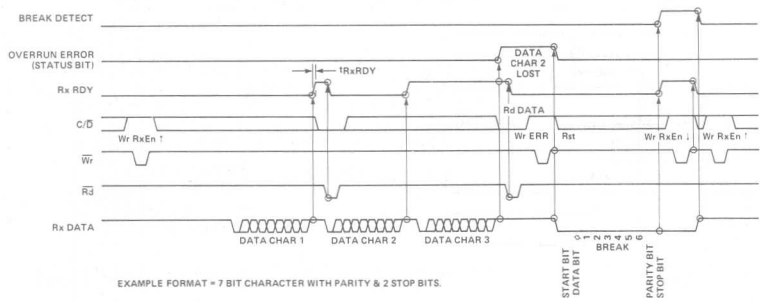
NOTE #1: T_{WC} INCLUDES THE RESPONSE TIMING OF A CONTROL BYTE.
NOTE #2: T_{CR} INCLUDES THE EFFECT OF CTS ON THE TxENBL CIRCUITRY.

Transmitter Control & Flag Timing (ASYNC Mode)



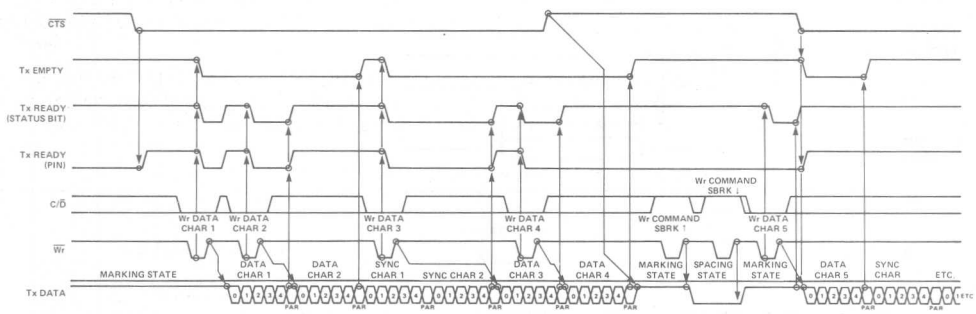
EXAMPLE FORMAT = 7 BIT CHARACTER WITH PARITY & 2 STOP BITS.

Receiver Control & Flag Timing (ASYNC Mode)



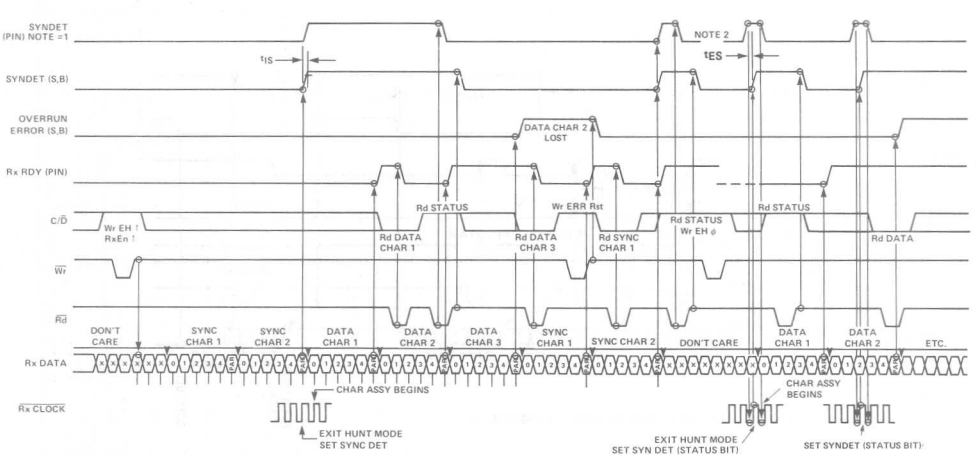
EXAMPLE FORMAT = 7 BIT CHARACTER WITH PARITY & 2 STOP BITS.

Transmitter Control & Flag Timing (SYNC Mode)



EXAMPLE FORMAT = 5 BIT CHARACTER WITH PARITY, 2 SYNC CHARACTERS.

Receiver Control & Flag Timing (SYNC Mode)



NOTE =1: INTERNAL SYNC, 2 SYNC CHARACTERS, 5 BITS, WITH PARITY
NOTE =2: EXTERNAL SYNC, 5 BITS, WITH PARITY

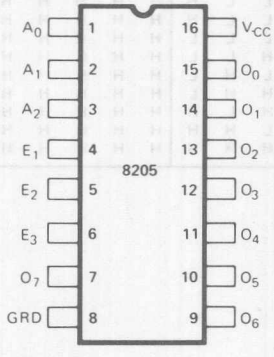
8205
HIGH SPEED 1 OUT OF 8 BINARY DECODER

- I/O Port or Memory Selector
- Simple Expansion — Enable Inputs
- High Speed Schottky Bipolar Technology — 18 ns Max Delay
- Directly Compatible with TTL Logic Circuits
- Low Input Load Current — 0.25 mA Max, 1/6 Standard TTL Input Load
- Minimum Line Reflection — Low Voltage Diode Input Clamp
- Outputs Sink 10 mA Min
- 16-Pin Dual In-Line Ceramic or Plastic Package

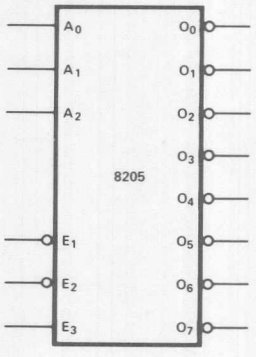
The Intel® 8205 decoder can be used for expansion of systems which utilize input ports, output ports, and memory components with active low chip select input. When the 8205 is enabled, one of its 8 outputs goes "low", thus a single row of a memory system is selected. The 3-chip enable inputs on the 8205 allow easy system expansion. For very large systems, 8205 decoders can be cascaded such that each decoder can drive 8 other decoders for arbitrary memory expansions.

The 8205 is packaged in a standard 16-pin dual in-line package, and its performance is specified over the temperature range of 0°C to +75°C, ambient. The use of Schottky barrier diode clamped transistors to obtain fast switching speeds results in higher performance than equivalent devices made with a gold diffusion process.

PIN CONFIGURATION



LOGIC SYMBOL



PIN NAMES

A_0-A_2	ADDRESS INPUTS
$\overline{E_1}$, $\overline{E_3}$	ENABLE INPUTS
$\overline{O_0}$-$\overline{O_7}$	DECODED OUTPUTS

ADDRESS			ENABLE			OUTPUTS							
A_0	A_1	A_2	E_1	E_2	E_3	0	1	2	3	4	5	6	7
L	L	L	L	L	H	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
L	H	L	L	L	H	H	H	L	H	H	H	H	H
H	H	L	L	L	H	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
H	L	H	L	L	H	H	H	H	H	H	L	H	H
L	H	H	L	L	H	H	H	H	H	H	H	L	H
H	H	H	L	L	H	H	H	H	H	H	H	H	L
X	X	X	L	L	L	H	H	H	H	H	H	H	H
X	X	X	H	L	L	H	H	H	H	H	H	H	H
X	X	X	L	H	L	H	H	H	H	H	H	H	H
X	X	X	H	H	L	H	H	H	H	H	H	H	H
X	X	X	H	L	H	H	H	H	H	H	H	H	H
X	X	X	L	H	H	H	H	H	H	H	H	H	H
X	X	X	H	H	H	H	H	H	H	H	H	H	H

FUNCTIONAL DESCRIPTION

Decoder

The 8205 contains a one out of eight binary decoder. It accepts a three bit binary code and by gating this input, creates an exclusive output that represents the value of the input code.

For example, if a binary code of 101 was present on the A0, A1 and A2 address input lines, and the device was enabled, an active low signal would appear on the $\overline{05}$ output line. Note that all of the other output pins are sitting at a logic high, thus the decoded output is said to be exclusive. The decoders outputs will follow the truth table shown below in the same manner for all other input variations.

Enable Gate

When using a decoder it is often necessary to gate the outputs with timing or enabling signals so that the exclusive output of the decoded value is synchronous with the overall system.

The 8205 has a built-in function for such gating. The three enable inputs ($\overline{E1}$, $\overline{E2}$, E3) are ANDed together and create a single enable signal for the decoder. The combination of both active "high" and active "low" device enable inputs provides the designer with a powerfully flexible gating function to help reduce package count in his system.

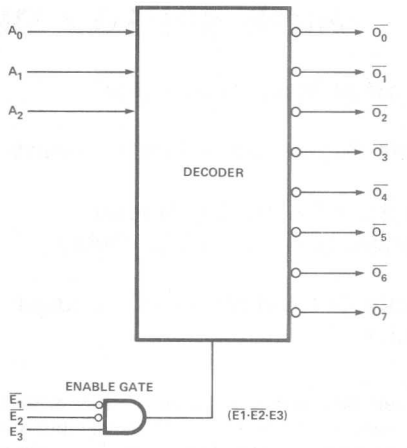


Figure 1. Enable Gate

ADDRESS			ENABLE			OUTPUTS							
A_0	A_1	A_2	E_1	E_2	E_3	0	1	2	3	4	5	6	7
L	L	L	L	L	H	L	H	H	H	H	H	H	H
H	L	L	L	L	H	H	L	H	H	H	H	H	H
L	H	L	L	L	H	H	H	L	H	H	H	H	H
H	H	L	L	L	H	H	H	H	L	H	H	H	H
L	L	H	L	L	H	H	H	H	H	L	H	H	H
H	L	H	L	L	H	H	H	H	H	H	L	H	H
L	H	H	L	L	H	H	H	H	H	H	H	L	H
H	H	H	L	L	H	H	H	H	H	H	H	H	L
X	X	X	L	L	L	H	H	H	H	H	H	H	H
X	X	X	H	L	L	H	H	H	H	H	H	H	H
X	X	X	L	H	L	H	H	H	H	H	H	H	H
X	X	X	H	H	L	H	H	H	H	H	H	H	H
X	X	X	H	L	H	H	H	H	H	H	H	H	H
X	X	X	L	H	H	H	H	H	H	H	H	H	H
X	X	X	H	H	H	H	H	H	H	H	H	H	H

APPLICATIONS OF THE 8205

The 8205 can be used in a wide variety of applications in microcomputer systems. I/O ports can be decoded from the address bus, chip select signals can be generated to select memory devices and the type of machine state such as in 8008 systems can be derived from a simple decoding of the state lines (S0, S1, S2) of the 8008 CPU.

I/O Port Decoder

Shown in the figure below is a typical application of the 8205. Address input lines are decoded by a group of 8205s (3). Each input has a binary weight. For example, A0 is assigned a value of 1 and is the LSB; A4 is assigned a value of 16 and is the MSB. By connecting them to the decoders as shown, an active low signal that is exclusive in nature and represents the value of the input address lines, is available at the outputs of the 8205s.

This circuit can be used to generate enable signals for I/O ports or any other decoder related application.

Note that no external gating is required to decode up to 24 exclusive devices and that a simple addition of an inverter or two will allow expansion to even larger decoder networks.

Chip Select Decoder

Using a very similar circuit to the I/O port decoder, an array of 8205s can be used to create a simple interface to a 24K memory system.

The memory devices used can be either ROM or RAM and are 1K in storage capacity. 8308s and 8102s are the devices typically used for this application. This type of memory device has ten (10) address inputs and an active "low" chip select ($\overline{CS}$). The lower order address bits A0-A9 which come from the microprocessor are "bussed" to all memory elements and the chip select to enable a specific device or group of devices comes from the array of 8205s. The output of the 8205 is active low so it is directly compatible with the memory components.

Basic operation is that the CPU issues an address to identify a specific memory location in which it wishes to "write" or "read" data. The most significant address bits A10-A14 are decoded by the array of 8205s and an exclusive, active low, chip select is generated that enables a specific memory device. The least significant address bits A0-A9 identify a specific location within the selected device. Thus, all addresses throughout the entire memory array are exclusive in nature and are non-redundant.

This technique can be expanded almost indefinitely to support even larger systems with the addition of a few inverters and an extra decoder (8205).

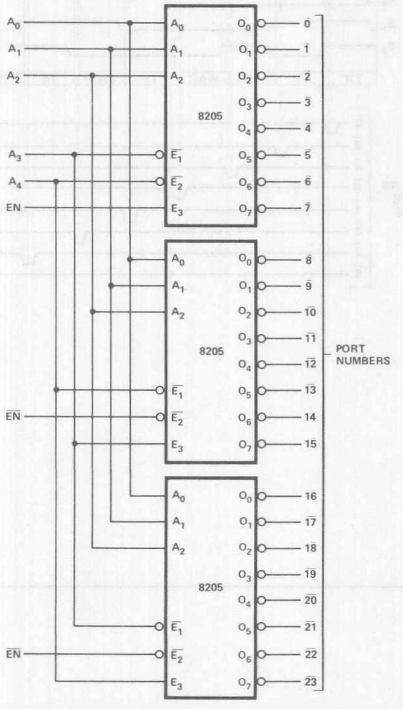


Figure 2. I/O Port Decoder

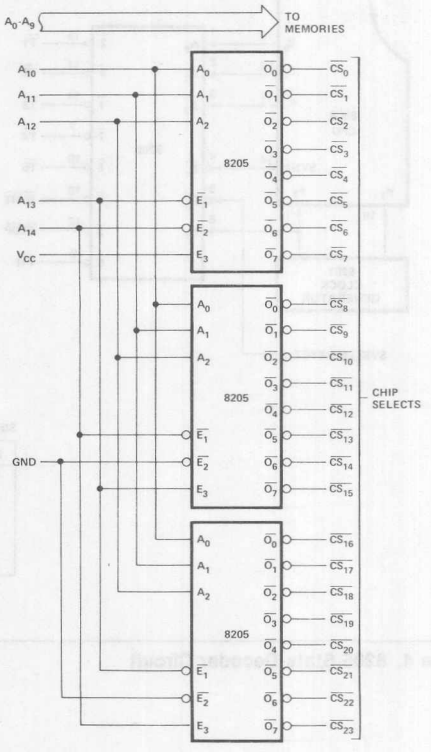


Figure 3. 32K Memory Interface

Logic Element Example

Probably the most overlooked application of the 8205 is that of a general purpose logic element. Using the "on-chip" enabling gate, the 8205 can be configured to gate its decoded outputs with system timing signals and generate strobes that can be directly connected to latches, flip-flops and one-shots that are used throughout the system.

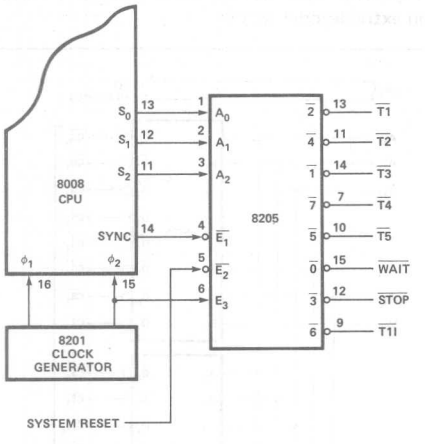
An excellent example of such an application is the "state decoder" in an 8008 CPU based system. The 8008 CPU issues three bits of information (S0, S1, S2) that indicate the nature of the data on the Data Bus during each machine state. Decoding of these signals is vital to generate strobes that can load the address latches, control bus discipline and general machine functions.

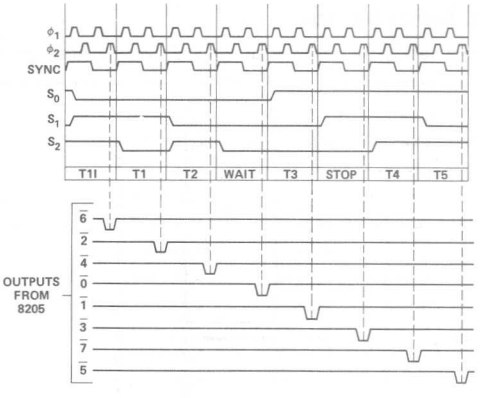
In the figure below a circuit is shown using the 8205 as the "state decoder" for an 8008 CPU that not only decodes the S0, S1, S2 outputs but gates these signals with the clock (phase 2) and the SYNC output of the 8008 CPU. The $\overline{T1}$ and $\overline{T2}$ decoded strobes can connect directly to devices like 8212s for latching the address information. The other decoded strobes can be used to generate signals to control the system data bus, memory timing functions and interrupt structure. RESET is connected to the enable gate so that strobes are not generated during system reset, eliminating accidental loading.

The power of such a circuit becomes evident when a single decoded strobe is logically broken down. Consider $\overline{T1}$ output, the boolean equation for it would be:

$$\overline{T1} = (\overline{S0} \cdot S1 \cdot \overline{S2}) \cdot (\overline{SYNC} \cdot \text{Phase } 2 \cdot \overline{\text{Reset}})$$

A six input NAND gate plus a few inverters would be needed to implement this function. The seven remaining outputs would need a similar circuit to duplicate their function, obviously a substantial savings in components can be achieved when using such a technique.





State Control Coding

S_0	S_1	S_2	STATE
0	1	0	T1
0	1	1	T1I
0	0	1	T2
0	0	0	WAIT
1	0	0	T3
1	1	0	STOP
1	1	1	T4
1	0	1	T5

Figure 4. 8205 State Decoder Circuit

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias:	Ceramic	−65°C to +125°C
	Plastic	−65°C to +75°C
Storage Temperature		−65°C to +160°C
All Output or Supply Voltages		−0.5 to +7 Volts
All Input Voltages		−1.0 to +5.5 Volts
Output Currents		125 mA

*COMMENT

Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. CHARACTERISTICS

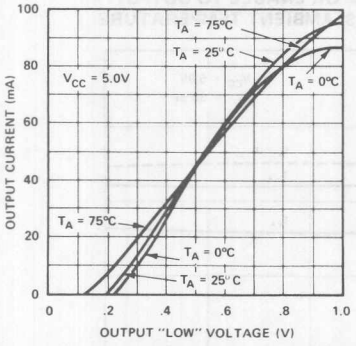
T_A = 0°C to +75°C, V_{CC} = 5V ±5%

8205

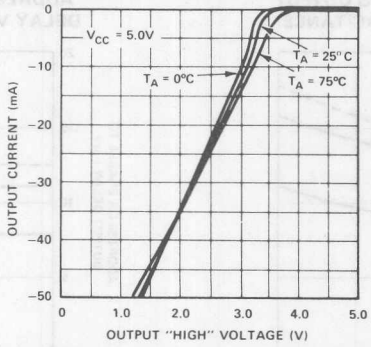
SYMBOL	PARAMETER	LIMIT MIN.	LIMIT MAX.	UNIT	TEST CONDITIONS
I_F	INPUT LOAD CURRENT		−0.25	mA	V_{CC} = 5.25V, V_F = 0.45V
I_R	INPUT LEAKAGE CURRENT		10	μA	V_{CC} = 5.25V, V_R = 5.25V
V_C	INPUT FORWARD CLAMP VOLTAGE		−1.0	V	V_{CC} = 4.75V, I_C = −5.0 mA
V_{OL}	OUTPUT "LOW" VOLTAGE		0.45	V	V_{CC} = 4.75V, I_{OL} = 10.0 mA
V_{OH}	OUTPUT HIGH VOLTAGE	2.4		V	V_{CC} = 4.75V, I_{OH} = −1.5 mA
V_{IL}	INPUT "LOW" VOLTAGE		0.85	V	V_{CC} = 5.0V
V_{IH}	INPUT "HIGH" VOLTAGE	2.0		V	V_{CC} = 5.0V
I_{SC}	OUTPUT HIGH SHORT CIRCUIT CURRENT	−40	−120	mA	V_{CC} = 5.0V, V_{OUT} = 0V
V_{OX}	OUTPUT "LOW" VOLTAGE @ HIGH CURRENT		0.8	V	V_{CC} = 5.0V, I_{OX} = 40 mA
I_{CC}	POWER SUPPLY CURRENT		70	mA	V_{CC} = 5.25V

TYPICAL CHARACTERISTICS

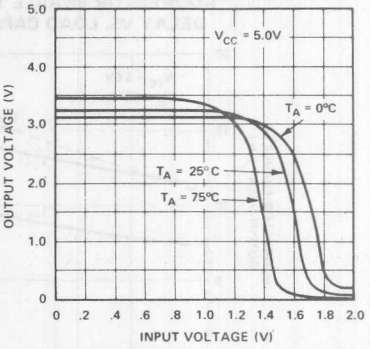
OUTPUT CURRENT VS. OUTPUT "LOW" VOLTAGE



OUTPUT CURRENT VS. OUTPUT "HIGH" VOLTAGE



DATA TRANSFER FUNCTION



SWITCHING CHARACTERISTICS

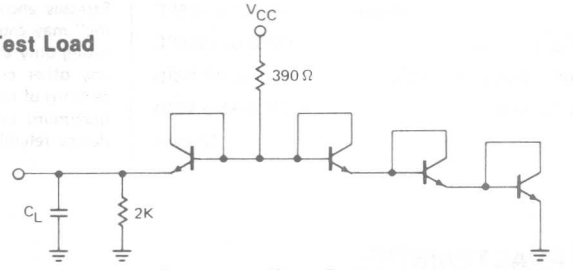
Conditions of Test:

Input pulse amplitudes: 2.5V

Input rise and fall times: 5 nsec between 1V and 2V

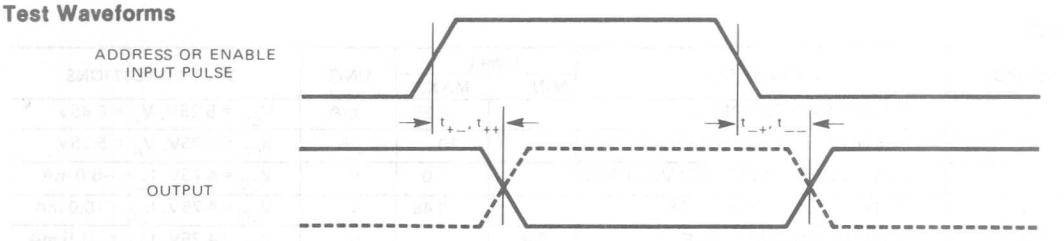
Measurements are made at 1.5V

Test Load



All Transistors 2N2369 or Equivalent. C_L = 30 pF

Test Waveforms



A.C. CHARACTRISTICS

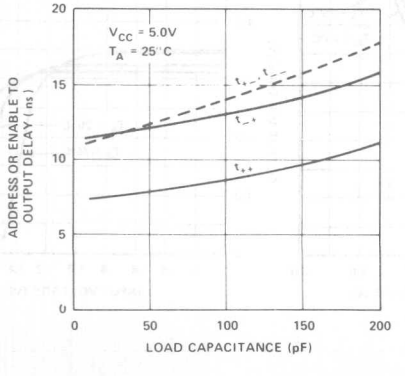
T_A = 0°C to +75°C, V_{CC} = 5V ±5% unless otherwise specified.

SYMBOL	PARAMETER		MAX. LIMIT	UNIT	TEST CONDITIONS
t_{++}	ADDRESS OR ENABLE TO OUTPUT DELAY		18	ns	
t_{-+}			18	ns	
t_{+-}			18	ns	
t_{--}			18	ns	
C_{IN} (1)	INPUT CAPACITANCE	P8205	4(typ.)	pF	f = 1 MHz, V_{CC} = 0V
		C8205	5(typ.)	pF	V_{BIAS} = 2.0V, T_A = 25°C

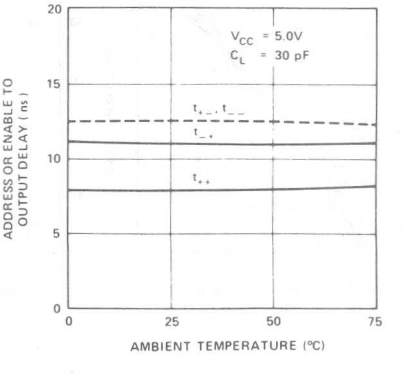
1. This parameter is periodically sampled and is not 100% tested.

TYPICAL CHARACTERISTICS

ADDRESS OR ENABLE TO OUTPUT DELAY VS. LOAD CAPACITANCE



ADDRESS OR ENABLE TO OUTPUT DELAY VS. AMBIENT TEMPERATURE



8214

PRIORITY INTERRUPT CONTROL UNIT

- **Eight Priority Levels**
- **Current Status Register**
- **Priority Comparator**
- **Fully Expandable**
- **High Performance (50ns)**
- **24-Pin Dual In-Line Package**

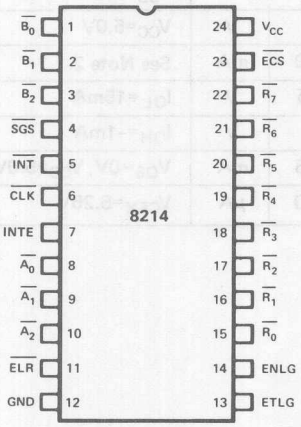
The 8214 is an eight level priority interrupt control unit designed to simplify interrupt driven microcomputer systems.

The PICU can accept eight requesting levels; determine the highest priority, compare this priority to a software controlled current status register and issue an interrupt to the system along with vector information to identify the service routine.

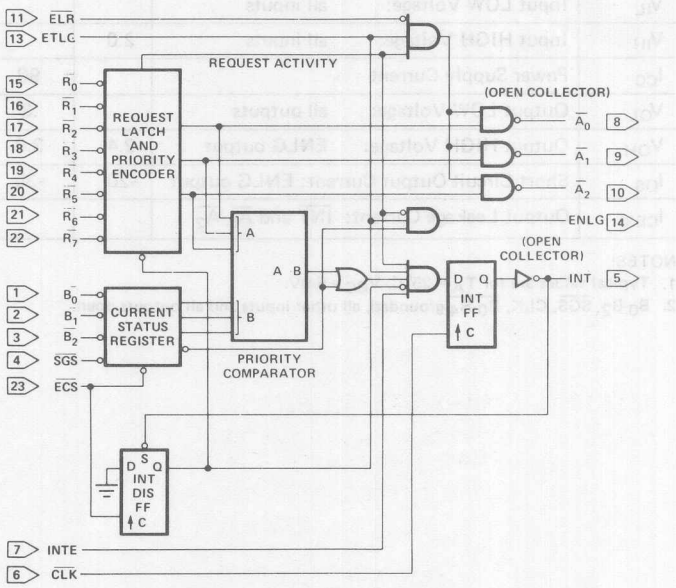
The 8214 is fully expandable by the use of open collector interrupt output and vector information. Control signals are also provided to simplify this function.

The PICU is designed to support a wide variety of vectored interrupt structures and reduce package count in interrupt driven microcomputer systems.

PIN CONFIGURATION



LOGIC DIAGRAM



PIN NAMES

INPUTS		
$\overline{R_0}$-$\overline{R_7}$	REQUEST LEVELS (R_7 HIGHEST PRIORITY)	
$\overline{B_0}$-$\overline{B_2}$	CURRENT STATUS	
$\overline{SGS}$	STATUS GROUP SELECT	
$\overline{ECS}$	ENABLE CURRENT STATUS	
INTE	INTERRUPT ENABLE	
$\overline{CLK}$	CLOCK (INT F-F)	
$\overline{ELR}$	ENABLE LEVEL READ	
ETLG	ENABLE THIS LEVEL GROUP	
OUTPUTS:		
$\overline{A_0}$-$\overline{A_2}$	REQUEST LEVELS	OPEN COLLECTOR
$\overline{INT}$	INTERRUPT (ACT. LOW)	OPEN COLLECTOR
ENLG	ENABLE NEXT LEVEL GROUP	

Temperature Under Bias 0°C to 70°C
Storage Temperature -65°C to +150°C
All Output and Supply Voltages -0.5V to +7V
All Input Voltages -1.0V to +5.5V
Output Currents 100 mA

*COMMENT: Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specifications is not implied. Exposure to absolute maximum ratings for extended periods may affect device reliability.

T_A = 0°C to +70°C, V_{CC} = 5V ±5%.

Symbol	Parameter	Limits			Unit	Conditions
		Min.	Typ.[1]	Max.		
V_C	Input Clamp Voltage (all inputs)			-1.0	V	I_C=-5mA
I_F	Input Forward Current: ETLG input all other inputs		-.15 -.08	-0.5 -0.25	mA mA	V_F=0.45V
I_R	Input Reverse Current: ETLG input all other inputs			80 40	μA μA	V_R=5.25V
V_{IL}	Input LOW Voltage: all inputs			0.8	V	V_{CC}=5.0V
V_{IH}	Input HIGH Voltage: all inputs	2.0			V	V_{CC}=5.0V
I_{CC}	Power Supply Current		90	130	mA	See Note 2.
V_{OL}	Output LOW Voltage: all outputs		.3	.45	V	I_{OL}=15mA
V_{OH}	Output HIGH Voltage: ENLG output	2.4	3.0		V	I_{OH}=-1mA
I_{OS}	Short Circuit Output Current: ENLG output	-20	-35	-55	mA	V_{OS}=0V, V_{CC}=5.0V
I_{CEX}	Output Leakage Current: $\overline{INT}$ and $\overline{A_0}$-$\overline{A_2}$			100	μA	V_{CEX}=5.25V

NOTES:
1. Typical values are for T_A = 25°C, V_{CC} = 5.0V.
2. B_0-B_2, $\overline{SGS}$, CLK, $\overline{R_0}$-$\overline{R_4}$ grounded, all other inputs and all outputs open.

A.C. CHARACTERISTICS AND WAVEFORMS $T_A = 0°C$ to $+70°C$, $V_{CC} = +5V \pm 5\%$

Symbol	Parameter	Limits Min.	Limits Typ.[1]	Limits Max.	Unit
t_{CY}	$\overline{CLK}$ Cycle Time	80	50		ns
t_{PW}	$\overline{CLK}$, $\overline{ECS}$, $\overline{INT}$ Pulse Width	25	15		ns
t_{ISS}	INTE Setup Time to $\overline{CLK}$	16	12		ns
t_{ISH}	INTE Hold Time after $\overline{CLK}$	20	10		ns
t_{ETCS}[2]	ETLG Setup Time to $\overline{CLK}$	25	12		ns
t_{ETCH}[2]	ETLG Hold Time After $\overline{CLK}$	20	10		ns
t_{ECCS}[2]	$\overline{ECS}$ Setup Time to $\overline{CLK}$	80	50		ns
t_{ECCH}[3]	$\overline{ECS}$ Hold Time After $\overline{CLK}$	0			ns
t_{ECRS}[3]	$\overline{ECS}$ Setup Time to $\overline{CLK}$	110	70		ns
t_{ECRH}[3]	$\overline{ECS}$ Hold Time After $\overline{CLK}$	0			
t_{ECSS}[2]	$\overline{ECS}$ Setup Time to $\overline{CLK}$	75	70		ns
t_{ECSH}[2]	$\overline{ECS}$ Hold Time After $\overline{CLK}$	0			ns
t_{DCS}[2]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Setup Time to $\overline{CLK}$	70	50		ns
t_{DCH}[2]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Hold Time After $\overline{CLK}$	0			ns
t_{RCS}[3]	$\overline{R_0}$-$\overline{R_7}$ Setup Time to $\overline{CLK}$	90	55		ns
t_{RCH}[3]	$\overline{R_0}$-$\overline{R_7}$ Hold Time After $\overline{CLK}$	0			ns
t_{ICS}	$\overline{INT}$ Setup Time to $\overline{CLK}$	55	35		ns
t_{CI}	$\overline{CLK}$ to $\overline{INT}$ Propagation Delay		15	25	ns
t_{RIS}[4]	$\overline{R_0}$-$\overline{R_7}$ Setup Time to $\overline{INT}$	10	0		ns
t_{RIH}[4]	$\overline{R_0}$-$\overline{R_7}$ Hold Time After $\overline{INT}$	35	20		ns
t_{RA}	$\overline{R_0}$-$\overline{R_7}$ to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		80	100	ns
t_{ELA}	$\overline{ELR}$ to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		40	55	ns
t_{ECA}	$\overline{ECS}$ to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		100	120	ns
t_{ETA}	ETLG to $\overline{A_0}$-$\overline{A_2}$ Propagation Delay		35	70	ns
t_{DECS}[4]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Setup Time to $\overline{ECS}$	15	10		ns
t_{DECH}[4]	$\overline{SGS}$ and $\overline{B_0}$-$\overline{B_2}$ Hold Time After $\overline{ECS}$	15	10		ns
t_{REN}	$\overline{R_0}$-$\overline{R_7}$ to ENLG Propagation Delay		45	70	ns
t_{ETEN}	ETLG to ENLG Propagation Delay		20	25	ns
t_{ECRN}	$\overline{ECS}$ to ENLG Propagation Delay		85	90	ns
t_{ECSN}	$\overline{ECS}$ to ENLG Propagation Delay		35	55	ns

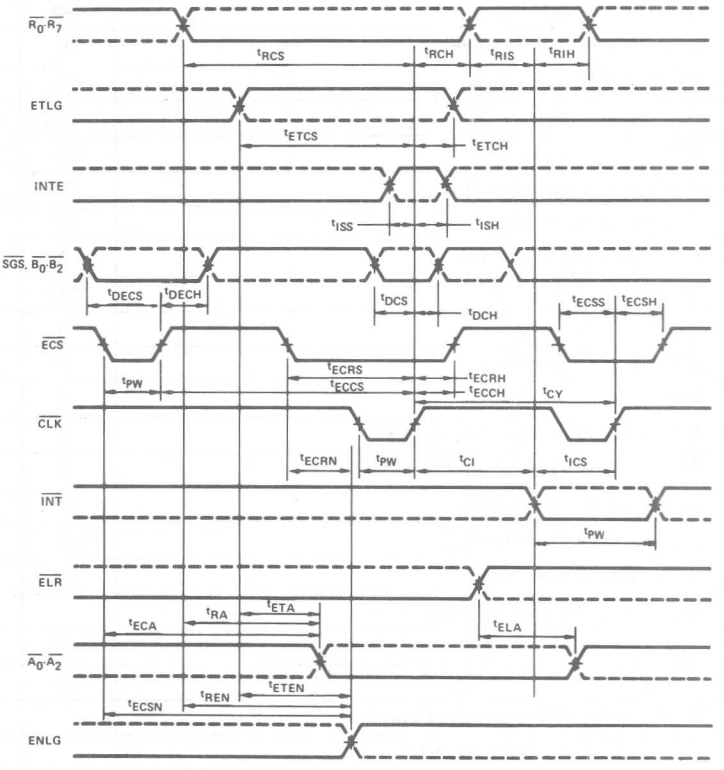
CAPACITANCE [5]

Symbol	Parameter	Limits Min.	Limits Typ.[1]	Limits Max	Unit
C_{IN}	Input Capacitance		5	10	pF
C_{OUT}	Output Capacitance		7	12	pF

TEST CONDITIONS: $V_{BIAS} = 2.5V$, $V_{CC} = 5V$, $T_A = 25°C$, f = 1 MHz

NOTE 5. This parameter is periodically sampled and not 100% tested.

WAVEFORMS



NOTES:

(1) Typical values are for $T_A = 25°C$, $V_{CC} = 5.0V$.

(2) Required for proper operation if ISE is enabled during next clock pulse.

(3) These times are not required for proper operation but for desired change in interrupt flip-flop.

(4) Required for new request or status to be properly loaded.

TEST CONDITIONS:

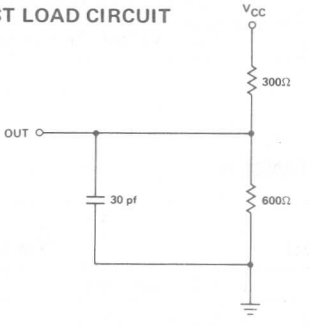
Input pulse amplitude: 2.5 volts.

Input rise and fall times: 5 ns between 1 and 2 volts.

Output loading of 15 mA and 30 pf.

Speed measurements taken at the 1.5V levels.

TEST LOAD CIRCUIT



4 BIT PARALLEL BIDIRECTIONAL BUS DRIVER

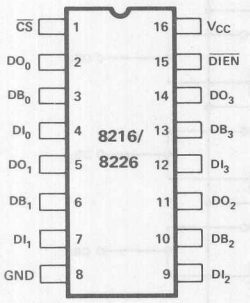
- **Data Bus Buffer Driver for 8080 CPU**
- **Low Input Load Current — .25 mA Maximum**
- **High Output Drive Capability for Driving System Data Bus**
- **3.65V Output High Voltage for Direct Interface to 8080 CPU**
- **Three State Outputs**
- **Reduces System Package Count**

The 8216/8226 is a 4-bit bi-directional bus driver/receiver.

All inputs are low power TTL compatible. For driving MOS, the DO outputs provide a high 3.65V V_{OH}, and for high capacitance terminated bus structures, the DB outputs provide a high 50mA I_{OL} capability.

A non-inverting (8216) and an inverting (8226) are available to meet a wide variety of applications for buffering in microcomputer systems.

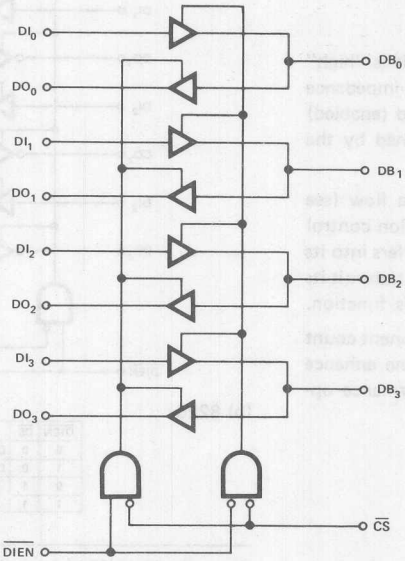
PIN CONFIGURATION



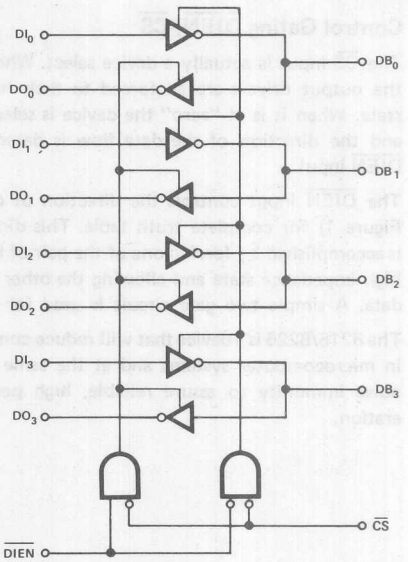
PIN NAMES

DB_0-DB_3	DATA BUS BI-DIRECTIONAL
DI_0-DI_3	DATA INPUT
DO_0-DO_3	DATA OUTPUT
$\overline{DIEN}$	DATA IN ENABLE DIRECTION CONTROL
$\overline{CS}$	CHIP SELECT

LOGIC DIAGRAM 8216



LOGIC DIAGRAM 8226



FUNCTIONAL DESCRIPTION

Microprocessors like the 8080 are MOS devices and are generally capable of driving a single TTL load. The same is true for MOS memory devices. While this type of drive is sufficient in small systems with few components, quite often it is necessary to buffer the microprocessor and memories when adding components or expanding to a multi-board system.

The 8216/8226 is a four bit bi-directional bus driver specifically designed to buffer microcomputer system components.

Bi-Directional Driver

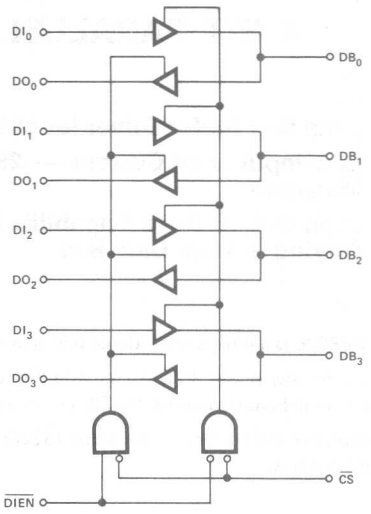
Each buffered line of the four bit driver consists of two separate buffers that are tri-state in nature to achieve direct bus interface and bi-directional capability. On one side of the driver the output of one buffer and the input of another are tied together (DB), this side is used to interface to the system side components such as memories, I/O, etc., because its interface is direct TTL compatible and it has high drive (50mA). On the other side of the driver the inputs and outputs are separated to provide maximum flexibility. Of course, they can be tied together so that the driver can be used to buffer a true bi-directional bus such as the 8080 Data Bus. The DO outputs on this side of the driver have a special high voltage output drive capability (3.65V) so that direct interface to the 8080 and 8008 CPUs is achieved with an adequate amount of noise immunity (350mV worst case).

Control Gating $\overline{DIEN}$, $\overline{CS}$

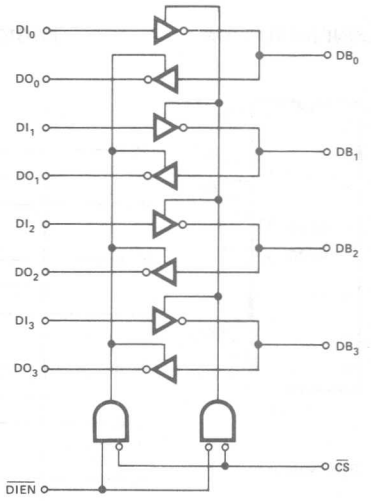
The $\overline{CS}$ input is actually a device select. When it is "high" the output drivers are all forced to their high-impedance state. When it is at "zero" the device is selected (enabled) and the direction of the data flow is determined by the $\overline{DIEN}$ input.

The $\overline{DIEN}$ input controls the direction of data flow (see Figure 1) for complete truth table. This direction control is accomplished by forcing one of the pair of buffers into its high impedance state and allowing the other to transmit its data. A simple two gate circuit is used for this function.

The 8216/8226 is a device that will reduce component count in microcomputer systems and at the same time enhance noise immunity to assure reliable, high performance operation.



(a) 8216



(b) 8226

$\overline{DIEN}$	$\overline{CS}$	
0	0	DI → DB
1	0	DB → DO
0	1	HIGH IMPEDANCE
1	1	

Figure 1. 8216/8226 Logic Diagrams

APPLICATIONS OF 8216/8226

8080 Data Bus Buffer

The 8080 CPU Data Bus is capable of driving a single TTL load and is more than adequate for small, single board systems. When expanding such a system to more than one board to increase I/O or Memory size, it is necessary to provide a buffer. The 8216/8226 is a device that is exactly fitted to this application.

Shown in Figure 2 are a pair of 8216/8226 connected directly to the 8080 Data Bus and associated control signals. The buffer is bi-directional in nature and serves to isolate the CPU data bus.

On the system side, the DB lines interface with standard semiconductor I/O and Memory components and are completely TTL compatible. The DB lines also provide a high drive capability (50mA) so that an extremely large system can be dirven along with possible bus termination networks.

On the 8080 side the DI and DO lines are tied together and are directly connected to the 8080 Data Bus for bi-directional operation. The DO outputs of the 8216/8226 have a high voltage output capability of 3.65 volts which allows direct connection to the 8080 whose minimum input voltage is 3.3 volts. It also gives a very adequate noise margin of 350mV (worst case).

The $\overline{DIEN}$ inputs to 8216/8226 is connected directly to the 8080. $\overline{DIEN}$ is tied to DBIN so that proper bus flow is maintained, and $\overline{CS}$ is tied to $\overline{BUSEN}$ so that the system side Data Bus will be 3-stated when a Hold request has been acknowledged during a DMA activity.

Memory and I/O Interface to a Bi-directional Bus

In large microcomputer systems it is often necessary to provide Memory and I/O with their own buffers and at the same time maintain a direct, common interface to a bi-directional Data Bus. The 8216/8226 has separated data in and data out lines on one side and a common bi-directional set on the other to accomodate such a function.

Shown in Figure 3 is an example of how the 8216/8226 is used in this type of application.

The interface to Memory is simple and direct. The memories used are typically Intel® 8102, 8102A, 8101 or 8107B-4 and have separate data inputs and outputs. The DI and DO lines of the 8216/8226 tie to them directly and under control of the $\overline{MEMR}$ signal, which is connected to the $\overline{DIEN}$ input, an interface to the bi-directional Data Bus is maintained.

The interface to I/O is similar to Memory. The I/O devices used are typically Intel® 8255s, and can be used for both input and output ports. The $\overline{I/O\ R}$ signal is connected directly to the $\overline{DIEN}$ input so that proper data flow from the I/O device to the Data Bus is maintained.

The 8216/8226 can be used in a wide variety of other buffering functions in microcomputer systems such as Address Bus Drivers, Drivers to peripheral devices such as printers, and as Drivers for long length cables to other peripherals or systems.

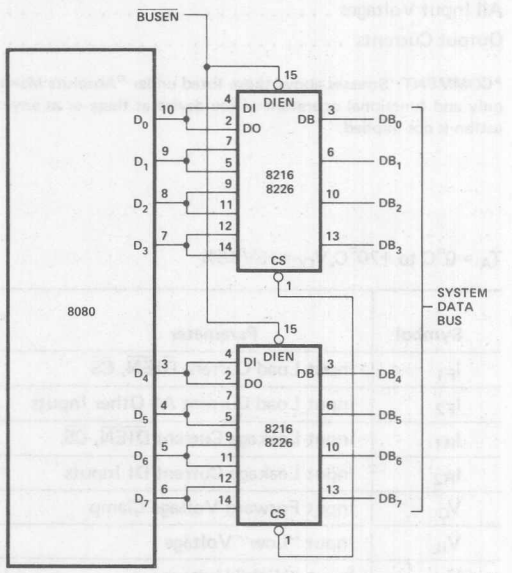


Figure 2. 8080 Data Bus Buffer.

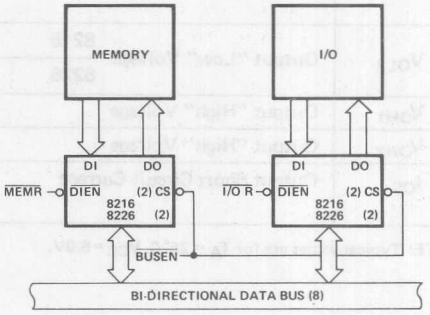


Figure 3. Memory and I/O Interface to a Bi-Directional Bus.

D.C. AND OPERATING CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS*

Temperature Under Bias 0°C to 70°C
Storage Temperature -65°C to +150°C
All Output and Supply Voltages -0.5V to +7V
All Input Voltages -1.0V to +5.5V
Output Currents 125 mA

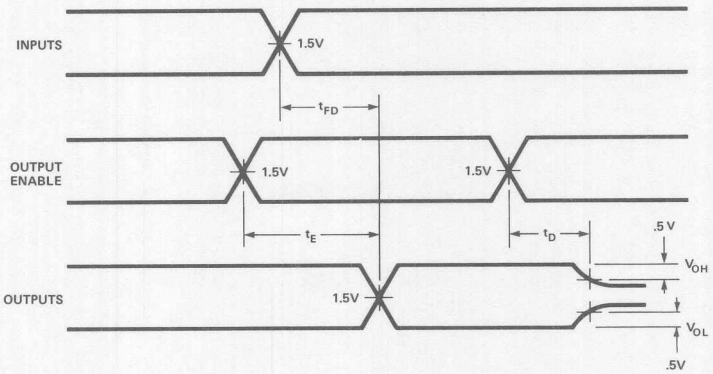
*COMMENT: Stresses above those listed under "Absolute Maximum Rating" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or at any other condition above those indicated in the operational sections of this specification is not implied.

T_A = 0°C to +70°C, V_{CC}=+5V ±5%

Symbol	Parameter		Limits Min.	Typ.	Max.	Unit	Conditions
I_{F1}	Input Load Current $\overline{DIEN}$, $\overline{CS}$			-0.15	-.5	mA	V_F = 0.45
I_{F2}	Input Load Current All Other Inputs			-0.08	-.25	mA	V_F = 0.45
I_{R1}	Input Leakage Current $\overline{DIEN}$, $\overline{CS}$				20	μA	V_R = 5.25V
I_{R2}	Input Leakage Current DI Inputs				10	μA	V_R = 5.25V
V_C	Input Forward Voltage Clamp				-1	V	I_C = -5mA
V_{IL}	Input "Low" Voltage				.95	V	
V_{IH}	Input "High" Voltage		2.0			V	
$\|I_O\|$	Output Leakage Current (3-State)	DO DB			20 100	μA	V_O = 0.45V/5.25V
I_{CC}	Power Supply Current	8216		95	130	mA	
		8226		85	120	mA	
V_{OL1}	Output "Low" Voltage			0.3	.45	V	DO Outputs I_{OL}=15mA DB Outputs I_{OL}=25mA
V_{OL2}	Output "Low" Voltage	8216		0.5	.6	V	DB Outputs I_{OL}=55mA
		8226		0.5	.6	V	DB Outputs I_{OL}=50mA
V_{OH1}	Output "High" Voltage		3.65	4.0		V	DO Outputs I_{OH} = -1mA
V_{OH2}	Output "High" Voltage		2.4	3.0		V	DB Outputs I_{OH} = -10mA
I_{OS}	Output Short Circuit Current		-15 -30	-35 -75	-65 -120	mA mA	DO Outputs $V_O \cong 0V$, DB Outputs V_{CC}=5.0V

NOTE: Typical values are for T_A = 25°C, V_{CC} = 5.0V.

WAVEFORMS



A.C. CHARACTERISTICS

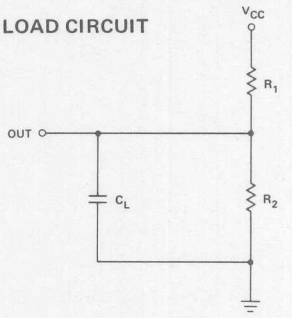
T_A = 0°C to +70°C, V_{CC} = +5V ±5%

Symbol	Parameter	Limits Min.	Typ.[1]	Max.	Unit	Conditions
T_{PD1}	Input to Output Delay DO Outputs		15	25	ns	C_L=30pF, R_1=300Ω R_2=600Ω
T_{PD2}	Input to Output Delay DB Outputs					
	8216		20	30	ns	C_L=300pF, R_1=90Ω
	8226		16	25	ns	R_2 = 180Ω
T_E	Output Enable Time					
	8216		45	65	ns	(Note 2)
	8226		35	54	ns	(Note 3)
T_D	Output Disable Time		20	35	ns	(Note 4)

TEST CONDITIONS:

Input pulse amplitude of 2.5V.
Input rise and fall times of 5 ns between 1 and 2 volts.
Output loading is 5 mA and 10 pF.
Speed measurements are made at 1.5 volt levels.

TEST LOAD CIRCUIT

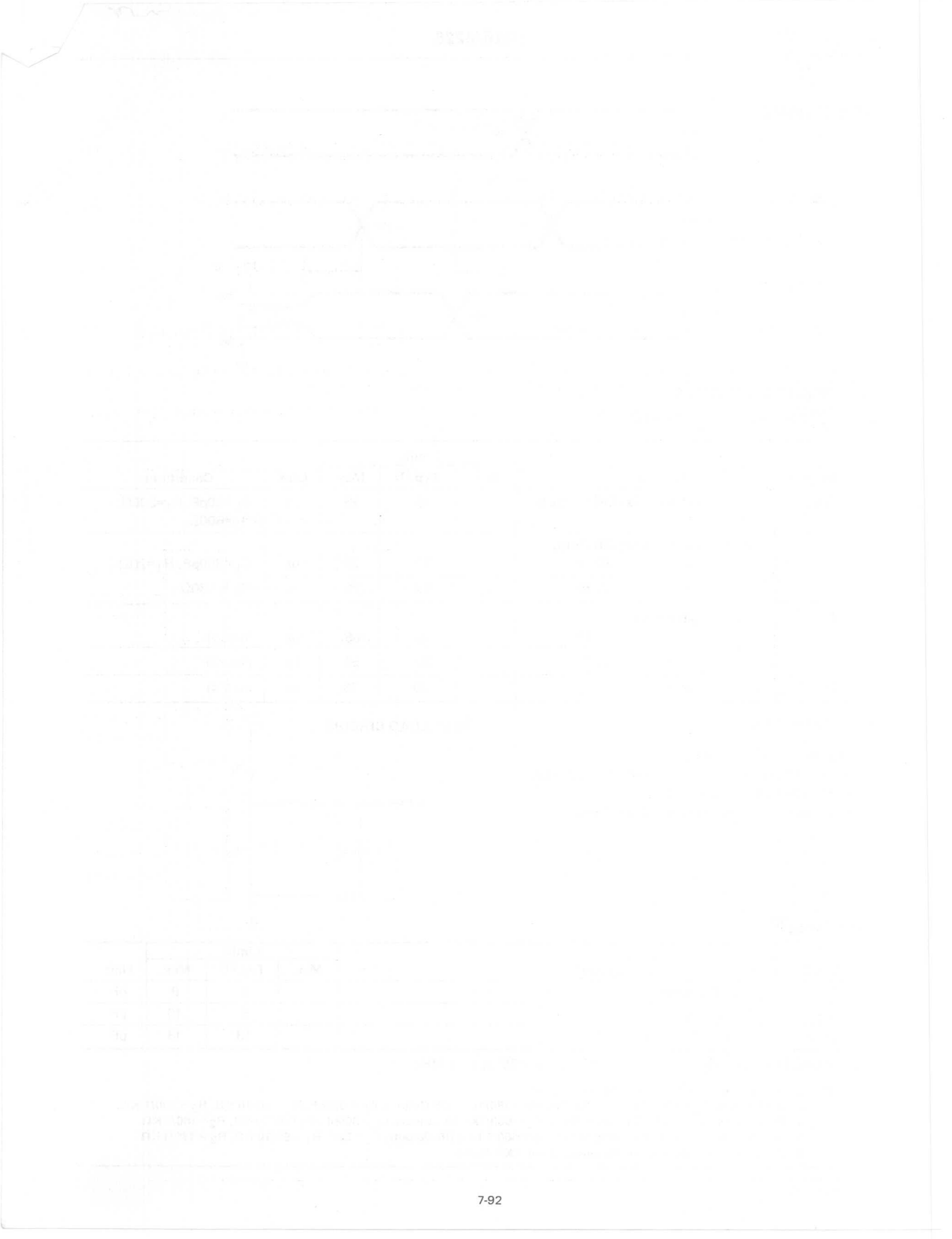


Capacitance[5]

Symbol	Parameter	Limits Min.	Typ.[1]	Max.	Unit
C_{IN}	Input Capacitance		4	8	pF
C_{OUT1}	Output Capacitance		6	10	pF
C_{OUT2}	Output Capacitance		13	18	pF

TEST CONDITIONS: V_{BIAS} = 2.5V, V_{CC} = 5.0V, T_A = 25°C, f = 1 MHz.

NOTES:
1. Typical values are for T_A = 25°C, V_{CC} = 5.0V.
2. DO Outputs, C_L = 30pF, R_1 = 300/10 KΩ, R_2 = 180/1KΩ; DB Outputs, C_L = 300pF, R_1 = 90/10 KΩ, R_2 = 180/1 KΩ.
3. DO Outputs, C_L = 30pF, R_1 = 300/10 KΩ, R_2 = 600/1K; DB Outputs, C_L = 300pF, R_1 = 90/10 KΩ, R_2 = 180/1 KΩ.
4. DO Outputs, C_L = 5pF, R_1 = 300/10 KΩ, R_2 = 600/1 KΩ; DB Outputs, C_L = 5pF, R_1 = 90/10 KΩ, R_2 = 180/1 KΩ.
5. This parameter is periodically sampled and not 100% tested.



PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

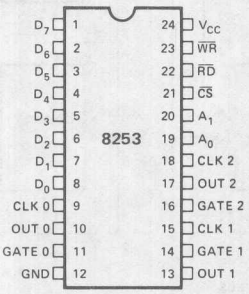
8253/8253-5
PROGRAMMABLE INTERVAL TIMER

- MCS—85™ Compatible 8253-5
- 3 Independent 16-Bit Counters
- DC to 2 MHz
- Programmable Counter Modes
- Count Binary or BCD
- Single +5V Supply
- 24-Pin Dual In-Line Package

The Intel® 8253 is a programmable counter/timer chip designed for use as an Intel microcomputer peripheral. It uses nMOS technology with a single +5V supply and is packaged in a 24-pin plastic DIP.

It is organized as 3 independent 16-bit counters, each with a count rate of up to 2 MHz. All modes of operation are software programmable.

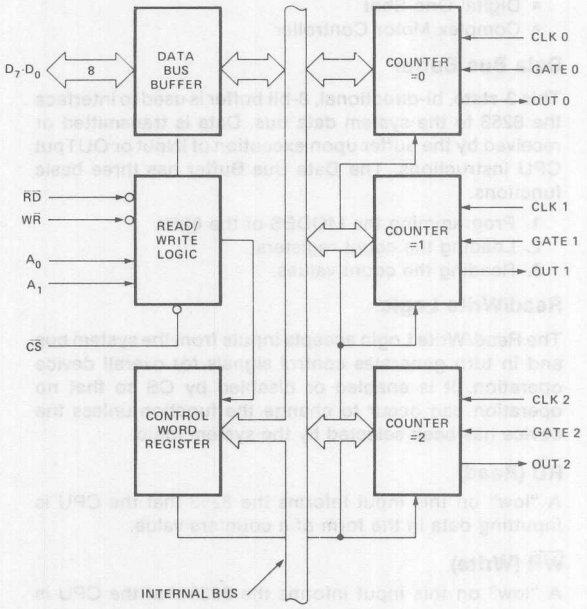
PIN CONFIGURATION



PIN NAMES

D_7-D_0	DATA BUS (8-BIT)
CLK N	COUNTER CLOCK INPUTS
GATE N	COUNTER GATE INPUTS
OUT N	COUNTER OUTPUTS
$\overline{RD}$	READ COUNTER
$\overline{WR}$	WRITE COMMAND OR DATA
$\overline{CS}$	CHIP SELECT
A_0-A_1	COUNTER SELECT
V_{CC}	+5 VOLTS
GND	GROUND

BLOCK DIAGRAM



© Intel Corporation, 1978

FUNCTIONAL DESCRIPTION

General

The 8253 is a programmable interval timer/counter specifically designed for use with the Intel™ Microcomputer systems. Its function is that of a general purpose, multi-timing element that can be treated as an array of I/O ports in the system software.

The 8253 solves one of the most common problems in any microcomputer system, the generation of accurate time delays under software control. Instead of setting up timing loops in systems software, the programmer configures the 8253 to match his requirements, initializes one of the counters of the 8253 with the desired quantity, then upon command the 8253 will count out the delay and interrupt the CPU when it has completed its tasks. It is easy to see that the software overhead is minimal and that multiple delays can easily be maintained by assignment of priority levels.

Other counter/timer functions that are non-delay in nature but also common to most microcomputers can be implemented with the 8253.

- Programmable Rate Generator
- Event Counter
- Binary Rate Multiplier
- Real Time Clock
- Digital One-Shot
- Complex Motor Controller

Data Bus Buffer

This 3-state, bi-directional, 8-bit buffer is used to interface the 8253 to the system data bus. Data is transmitted or received by the buffer upon execution of INput or OUTput CPU instructions. The Data Bus Buffer has three basic functions.

1. Programming the MODES of the 8253.
2. Loading the count registers.
3. Reading the count values.

Read/Write Logic

The Read/Write Logic accepts inputs from the system bus and in turn generates control signals for overall device operation. It is enabled or disabled by CS so that no operation can occur to change the function unless the device has been selected by the system logic.

$\overline{RD}$ (Read)

A "low" on this input informs the 8253 that the CPU is inputting data in the form of a counters value.

$\overline{WR}$ (Write)

A "low" on this input informs the 8253 that the CPU is outputting data in the form of mode information or loading counters.

A0, A1

These inputs are normally connected to the address bus. Their function is to select one of the three counters to be operated on and to address the control word register for mode selection.

$\overline{CS}$ (Chip Select)

A "low" on this input enables the 8253. No reading or writing will occur unless the device is selected. The $\overline{CS}$ input has no effect upon the actual operation of the counters.

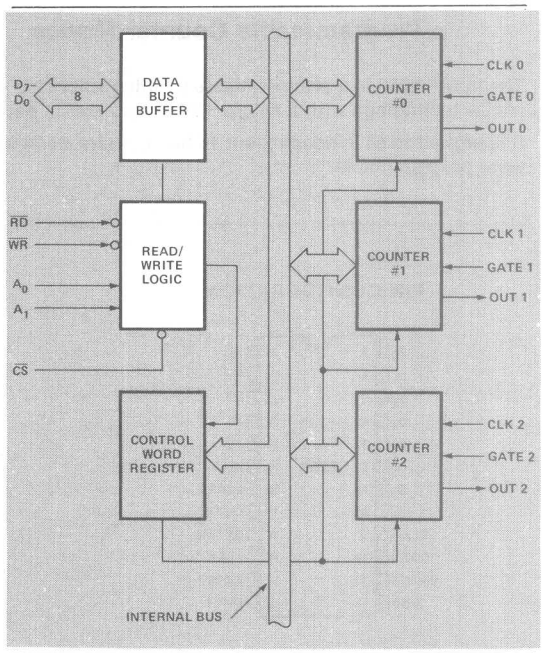


Figure 1. Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

$\overline{CS}$	$\overline{RD}$	$\overline{WR}$	A_1	A_0	
0	1	0	0	0	Load Counter No. 0
0	1	0	0	1	Load Counter No. 1
0	1	0	1	0	Load Counter No. 2
0	1	0	1	1	Write Mode Word
0	0	1	0	0	Read Counter No. 0
0	0	1	0	1	Read Counter No. 1
0	0	1	1	0	Read Counter No. 2
0	0	1	1	1	No-Operation 3-State
1	X	X	X	X	Disable 3-State
0	1	1	X	X	No-Operation 3-State

Control Word Register

The Control Word Register is selected when A0, A1 are 11. It then accepts information from the data bus buffer and stores it in a register. The information stored in this register controls the operational MODE of each counter, selection of binary or BCD counting and the loading of each count register.

The Control Word Register can only be written into; no read operation of its contents is available.

Counter #0, Counter #1, Counter #2

These three functional blocks are identical in operation so only a single Counter will be described. Each Counter consists of a single, 16-bit, pre-settable, DOWN counter. The counter can operate in either binary or BCD and its input, gate and output are configured by the selection of MODES stored in the Control Word Register.

The counters are fully independent and each can have separate Mode configuration and counting operation, binary or BCD. Also, there are special features in the control word that handle the loading of the count value so that software overhead can be minimized for these functions.

The reading of the contents of each counter is available to the programmer with simple READ operations for event counting applications and special commands and logic are included in the 8253 so that the contents of each counter can be read "on the fly" without having to inhibit the clock input.

8253 SYSTEM INTERFACE

The 8253 is a component of the Intel™ Microcomputer Systems and interfaces in the same manner as all other peripherals of the family. It is treated by the systems software as an array of peripheral I/O ports; three are counters and the fourth is a control register for MODE programming.

Basically, the select inputs A0, A1 connect to the A0, A1 address bus signals of the CPU. The $\overline{CS}$ can be derived directly from the address bus using a linear select method. Or it can be connected to the output of a decoder, such as an Intel® 8205 for larger systems.

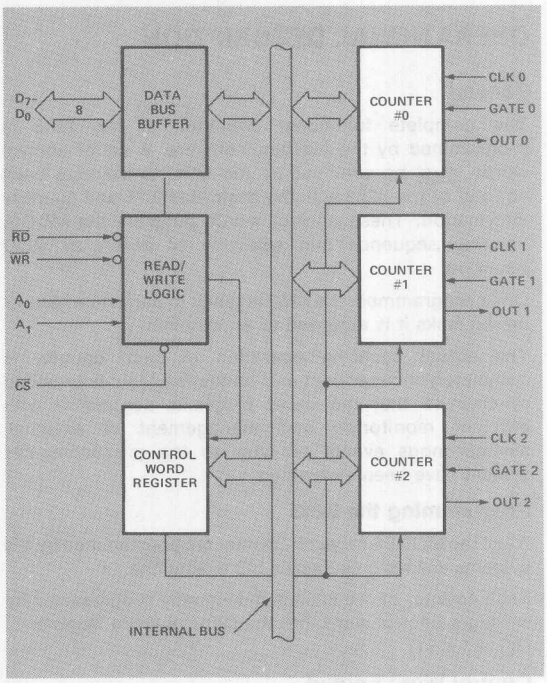


Figure 2. Block Diagram Showing Control Word Register and Counter Functions

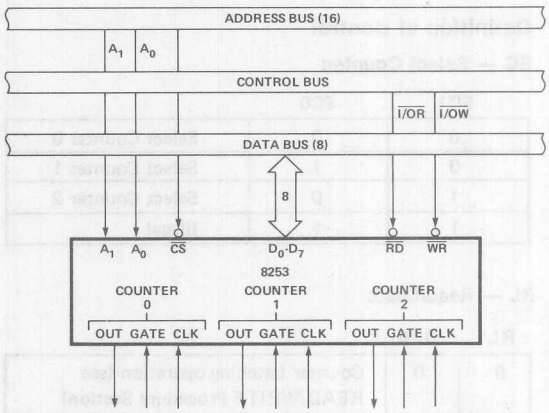


Figure 3. 8253 System Interface

OPERATIONAL DESCRIPTION

General

The complete functional definition of the 8253 is programmed by the systems software. A set of control words must be sent out by the CPU to initialize each counter of the 8253 with the desired MODE and quantity information. These control words program the MODE, Loading sequence and selection of binary or BCD counting.

Once programmed, the 8253 is ready to perform whatever timing tasks it is assigned to accomplish.

The actual counting operation of each counter is completely independent and additional logic is provided on-chip so that the usual problems associated with efficient monitoring and management of external, asynchronous events or rates to the microcomputer system have been eliminated.

Programming the 8253

All of the MODES for each counter are programmed by the systems software by simple I/O operations.

Each counter of the 8253 is individually programmed by writing a control word into the Control Word Register. (A0, A1 = 11)

Control Word Format

D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0
SC1	SC0	RL1	RL0	M2	M1	M0	BCD

Definition of Control

SC — Select Counter:

SC1	SC0	
0	0	Select Counter 0
0	1	Select Counter 1
1	0	Select Counter 2
1	1	Illegal

RL — Read/Load:

RL1	RL0	
0	0	Counter Latching operation (see READ/WRITE Procedure Section)
1	0	Read/Load most significant byte only.
0	1	Read/Load least significant byte only.
1	1	Read/Load least significant byte first, then most significant byte.

M — MODE:

M2	M1	M0	
0	0	0	Mode 0
0	0	1	Mode 1
X	1	0	Mode 2
X	1	1	Mode 3
1	0	0	Mode 4
1	0	1	Mode 5

BCD:

0	Binary Counter 16-bits
1	Binary Coded Decimal (BCD) Counter (4 Decades)

MODE Definition

MODE: Interrupt on Terminal Count. The output will be initially low after the mode set operation. After the count is loaded into the selected count register, the output will remain low and the counter will count. When terminal count is reached the output will go high and remain high until the selected count register is reloaded with the mode or a new count is loaded.

Reloading a counter register during counting results in the following:

(1) Load 1st byte stops the current counting.

(2) Load 2nd byte starts the new count.

MODE 1: Programmable One-Shot. The output will go low on the count following the rising edge of the gate input.

The output will go high on the terminal count. If a new count value is loaded while the output is low it will not affect the duration of the one-shot pulse until the succeeding trigger. The current count can be read at any time without affecting the one-shot pulse.

The one-shot is retriggerable, hence the output will remain low for the full count after any rising edge of the gate input.

MODE 2: Rate Generator. Divide by N counter. The output will be low for one period of the input clock. The period from one output pulse to the next equals the number of input counts in the count register. If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value.

The gate input, when low, will force the output high. When the gate input goes high, the counter will start from the initial count. Thus, the gate input can be used to synchronize the counter.

When this mode is set, the output will remain high until after the count register is loaded. The output then can also be synchronized by software.

MODE 3: Square Wave Rate Generator. Similar to MODE 2 except that the output will remain high until one half the count has been completed (for even numbers) and go low for the other half of the count. If the count is odd, the output will be high for (N + 1)/2 counts and low for (N – 1)/2 counts.

If the counter register is reloaded with a new value during counting, this new value will be reflected immediately after after the output transition of the current count.

MODE 4: Software Triggered Strobe. After the mode is set, the output will be high. When the count is loaded, the counter will begin counting. On terminal count, the output will go low for one input clock period, then will go high again.

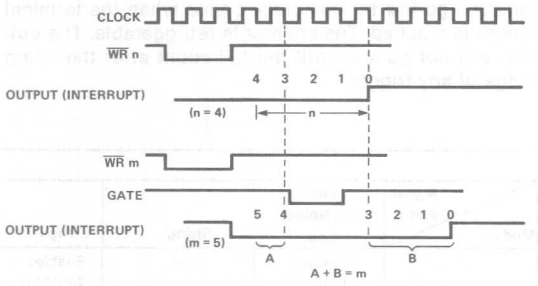
If the count register is reloaded between output pulses the present period will not be affected, but the subsequent period will reflect the new value. The count will be inhibited while the gate input is low. Reloading the counter register will restart counting beginning with the new number.

MODE 5: Hardware Triggered Strobe. The counter will start counting after the rising edge of the trigger input and will go low for one clock period when the terminal count is reached. The counter is retriggerable. The output will not go low until the full count after the rising edge of any trigger.

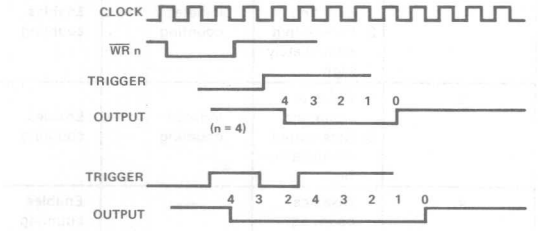
Modes \ Signal Status	Low Or Going Low	Rising	High
0	Disables counting	—	Enables counting
1	—	1) Initiates counting 2) Resets output after next clock	—
2	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
3	1) Disables counting 2) Sets output immediately high	Initiates counting	Enables counting
4	Disables counting	—	Enables counting
5	—	Initiates counting	—

Figure 4. Gate Pin Operations Summary

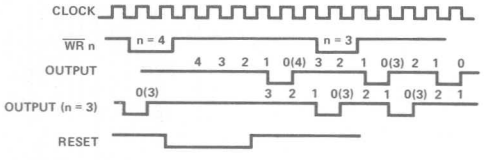
MODE 0



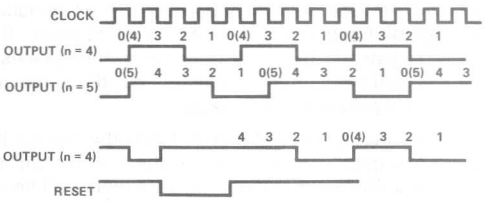
MODE 1



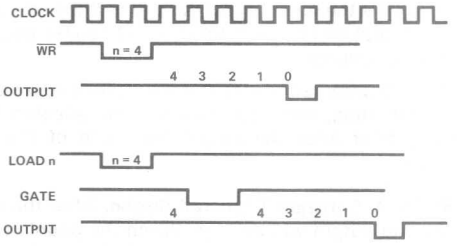
MODE 2



MODE 3



MODE 4



MODE 5

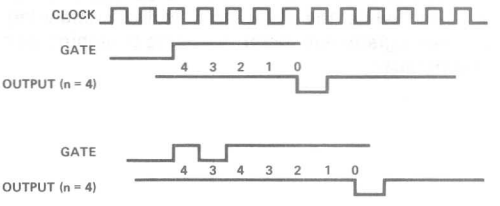


Figure 5. 8253 Timing Diagrams

8253 READ/WRITE PROCEDURE

Write Operations

The systems software must program each counter of the 8253 with the mode and quantity desired. The programmer must write out to the 8253 a MODE control word and the programmed number of count register bytes (1 or 2) prior to actually using the selected counter.

The actual order of the programming is quite flexible. Writing out of the MODE control word can be in any sequence of counter selection, e.g., counter #0 does not have to be first or counter #2 last. Each counter's MODE control word register has a separate address so that its loading is completely sequence independent. (SC0, SC1)

The loading of the Count Register with the actual count value, however, must be done in exactly the sequence programmed in the MODE control word (RL0, RL1). This loading of the counter's count register is still sequence independent like the MODE control word loading, but when a selected count register is to be loaded it must be loaded with the number of bytes programmed in the MODE control word (RL0, RL1). The one or two bytes to be loaded in the count register do not have to follow the associated MODE control word. They can be programmed at any time following the MODE control word loading as long as the correct number of bytes is loaded in order.

All counters are down counters. Thus, the value loaded into the count register will actually be decremented. Loading all zeroes into a count register will result in the maximum count (2^{16} for Binary or 10^4 for BCD). In MODE 0 the new count will not restart until the load has been completed. It will accept one of two bytes depending on how the MODE control words (RL0, RL1) are programmed. Then proceed with the restart operation.

	MODE Control Word Counter n
LSB	Count Register byte Counter n
MSB	Count Register byte Counter n

Note: Format shown is a simple example of loading the 8253 and does not imply that it is the only format that can be used.

Figure 6. Programming Format

			A1	A0
No. 1		MODE Control Word Counter 0	1	1
No. 2		MODE Control Word Counter 1	1	1
No. 3		MODE Control Word Counter 2	1	1
No. 4	LSB	Count Register Byte Counter 1	0	1
No. 5	MSB	Count Register Byte Counter 1	0	1
No. 6	LSB	Count Register Byte Counter 2	1	0
No. 7	MSB	Count Register Byte Counter 2	1	0
No. 8	LSB	Count Register Byte Counter 0	0	0
No. 9	MSB	Count Register Byte Counter 0	0	0

Note: The exclusive addresses of each counter's count register make the task of programming the 8253 a very simple matter, and maximum effective use of the device will result if this feature is fully utilized.

Figure 7. Alternate Programming Formats

Read Operations

In most counter applications it becomes necessary to read the value of the count in progress and make a computational decision based on this quantity. Event counters are probably the most common application that uses this function. The 8253 contains logic that will allow the programmer to easily read the contents of any of the three counters without disturbing the actual count in progress.

There are two methods that the programmer can use to read the value of the counters. The first method involves the use of simple I/O read operations of the selected counter. By controlling the A0, A1 inputs to the 8253 the programmer can select the counter to be read (remember that no read operation of the mode register is allowed A0, A1-11). The only requirement with this method is that in order to assure a stable count reading the actual operation of the selected counter must be inhibited either by controlling the Gate input or by external logic that inhibits the clock input. The contents of the counter selected will be available as follows:

first I/O Read contains the least significant byte (LSB).

second I/O Read contains the most significant byte (MSB).

Due to the internal logic of the 8253 it is absolutely necessary to complete the entire reading procedure. If two bytes are programmed to be read then two bytes must be read before any loading WR command can be sent to the same counter.

Read Operation Chart

A1	A0	RD	
0	0	0	Read Counter No. 0
0	1	0	Read Counter No. 1
1	0	0	Read Counter No. 2
1	1	0	Illegal

Reading While Counting

In order for the programmer to read the contents of any counter without effecting or disturbing the counting operation the 8253 has special internal logic that can be accessed using simple WR commands to the MODE register. Basically, when the programmer wishes to read the contents of a selected counter "on the fly" he loads the MODE register with a special code which latches the present count value into a storage register so that its contents contain an accurate, stable quantity. The programmer then issues a normal read command to the selected counter and the contents of the latched register is available.

MODE Register for Latching Count

A0, A1 = 11

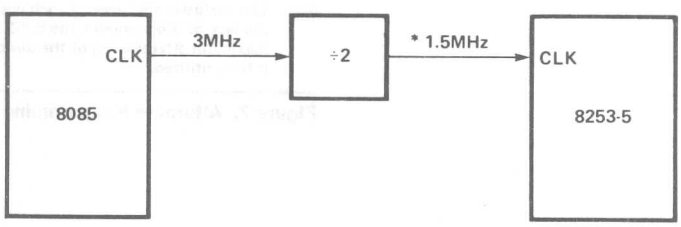
D7	D6	D5	D4	D3	D2	D1	D0
SC1	SC0	0	0	X	X	X	X

SC1,SC0 — specify counter to be latched.

D5,D4 — 00 designates counter latching operation.

X — don't care.

The same limitation applies to this mode of reading the counter as the previous method. That is, it is mandatory to complete the entire read operation as programmed.



*If an 8085 clock output is to drive an 8253-5 clock input, it must be reduced to 2 MHz or less.

Figure 8. MCS-85™ Clock Interface*

PRELIMINARY
Notice: This is a product in development. Some parameters may be subject to change.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias 0°C to 70°C
Storage Temperature −65°C to +150°C
Voltage On Any Pin
With Respect to Ground −0.5 V to +7 V
Power Dissipation 1 Watt

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS (T_A = 0°C to 70°C; V_{CC} = 5V ±5%)

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-0.5	0.8	V	
V_{IH}	Input High Voltage	2.2	V_{CC}+.5V	V	
V_{OL}	Output Low Voltage		0.45	V	Note 1
V_{OH}	Output High Voltage	2.4		V	Note 2
I_{IL}	Input Load Current		±10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = V_{CC} to 0V
I_{CC}	V_{CC} Supply Current		140	mA	

Note 1: 8253, I_{OL} = 1.6 mA; 8253-5, I_{OL} = 2.2 mA.
Note 2: 8253, I_{OH} = −150 μA; 8253-5, I_{OH} = −400 μA.

CAPACITANCE T_A = 25°C; V_{CC} = GND = 0V

Symbol	Parameter	Min.	Typ.	Max.	Unit	Test Conditions
C_{IN}	Input Capacitance			10	pF	fc = 1 MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to V_{SS}

PRELIMINARY final specification. Some are subject to change.

Bus Parameters (Note 1)

Read Cycle:

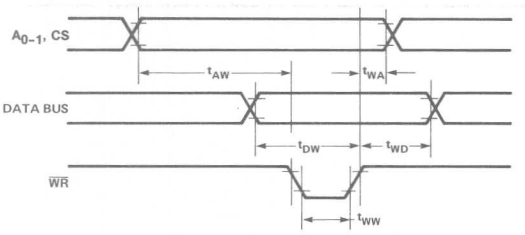
SYMBOL	PARAMETER	8253 MIN.	8253 MAX.	8253-5 MIN.	8253-5 MAX.	UNIT
t_{AR}	Address Stable Before $\overline{READ}$	50		50		ns
t_{RA}	Address Hold Time for $\overline{READ}$	5		5		ns
t_{RR}	$\overline{READ}$ Pulse Width	400		300		ns
t_{RD}	Data Delay From $\overline{READ}$[2]		300		200	ns
t_{DF}	$\overline{READ}$ to Data Floating	25	125	25	100	ns

Write Cycle:

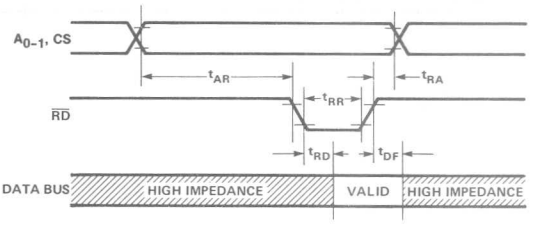
SYMBOL	PARAMETER	8253 MIN.	8253 MAX.	8253-5 MIN.	8253-5 MAX.	UNIT
t_{AW}	Address Stable Before $\overline{WRITE}$	50		50		ns
t_{WA}	Address Hold Time for $\overline{WRITE}$	30		30		ns
t_{WW}	$\overline{WRITE}$ Pulse Width	400		300		ns
t_{DW}	Data Set Up Time for $\overline{WRITE}$	300		250		ns
t_{WD}	Data Hold Time for $\overline{WRITE}$	40		30		ns
t_{RV}	Recovery Time Between $\overline{WRITES}$	1		1		μs

Notes: 1. AC timings measured at V_{OH} = 2.2, V_{OL} = 0.8
2. Test Conditions: 8253, C_L = 100pF; 8253-5: C_L = 150pF.

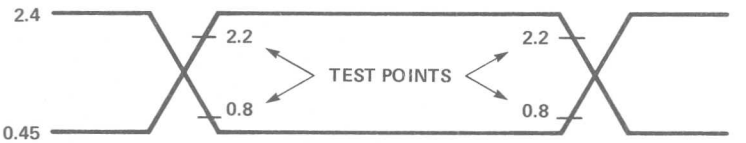
Write Timing:



Read Timing:



Input Waveforms for A.C. Tests:

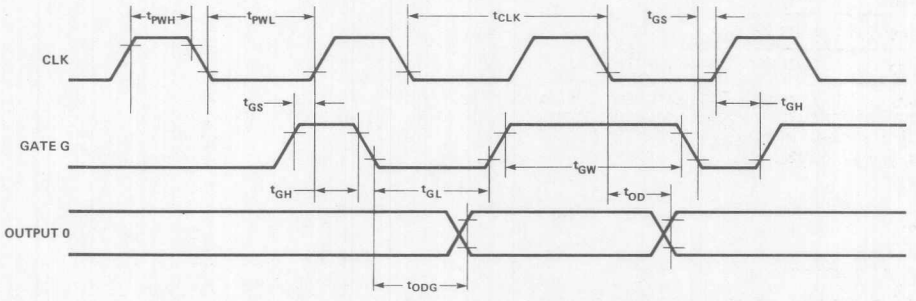


PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

Clock and Gate Timing:

SYMBOL	PARAMETER	8253		8253-5		UNIT
		MIN.	MAX.	MIN.	MAX.	
t_{CLK}	Clock Period	380	dc	380	dc	ns
t_{PWH}	High Pulse Width	230		230		ns
t_{PWL}	Low Pulse Width	150		150		ns
t_{GW}	Gate Width High	150		150		ns
t_{GL}	Gate Width Low	100		100		ns
t_{GS}	Gate Set Up Time to CLK↑	100		100		ns
t_{GH}	Gate Hold Time After CLK↑	50		50		ns
t_{OD}	Output Delay From CLK↓[1]		400		400	ns
t_{ODG}	Output Delay From Gate↓[1]		300		300	ns

Note 1: Test Conditions: 8253: C_L = 100pF; 8253-5: C_L = 150pF.



PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

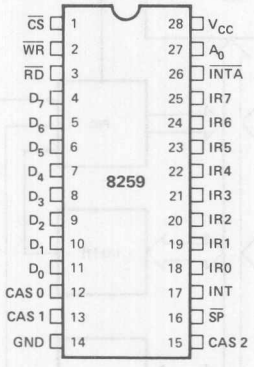
8259/8259-5
PROGRAMMABLE INTERRUPT CONTROLLER

- MCS-85™ Compatible 8259-5
- 8-Level Priority Controller
- Expandable to 64 Levels
- Programmable Interrupt Modes
- Individual Request Mask Capability
- Single +5V Supply (No Clocks)
- 28-Pin Dual In-Line Package

The Intel® 8259 handles up to 8 vectored priority interrupts for the CPU. It is cascadable for up to 64 vectored priority interrupts, without additional circuitry. It will be packaged in a 28-pin plastic DIP, uses nMOS technology and requires a single +5V supply. Circuitry is static, requiring no clock input.

The 8259 is designed to minimize the software and real time overhead in handling multi-level priority interrupts. It has several modes, permitting optimization for a variety of system requirements.

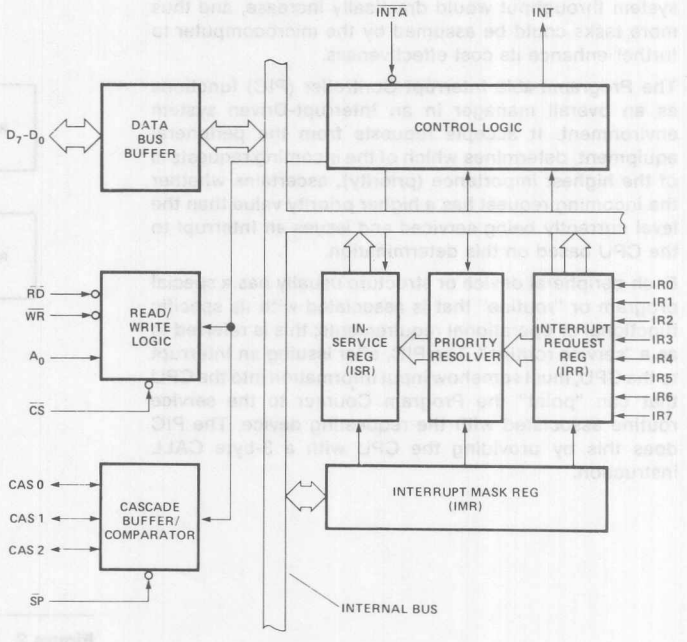
PIN CONFIGURATION



PIN NAMES

D_7-D_0	DATA BUS (BI-DIRECTIONAL)
$\overline{RD}$	READ INPUT
$\overline{WR}$	WRITE INPUT
A_0	COMMAND SELECT ADDRESS
$\overline{CS}$	CHIP SELECT
CAS1-CAS0	CASCADE LINES
$\overline{SP}$	SLAVE PROGRAM INPUT
INT	INTERRUPT OUTPUT
$\overline{INTA}$	INTERRUPT ACKNOWLEDGE INPUT
IR0-IR7	INTERRUPT REQUEST INPUTS

BLOCK DIAGRAM



© Intel Corporation, 1978

INTRODUCTION TO THE USE OF INTERRUPTS IN MICROCOMPUTER SYSTEMS

Microcomputer system design requires that I/O devices such as keyboards, displays, sensors and other components receive servicing in an efficient method so that large amounts of the total system tasks can be assumed by the microcomputer with little or no effect on throughput.

The most common method of servicing such devices is the **Polled** approach. This is where the processor must test each device in sequence and in effect "ask" each one if it needs servicing. It is easy to see that a large portion of the main program is looping through this continuence polling cycle and that such a method would have a serious, detrimental effect on system throughput thus limiting the tasks that could be assumed by the microcomputer and reducing the cost effectiveness of using such devices.

A more desireable method would be one that would allow the microprocessor to be executing its main program and only stop to service peripheral devices when it is told to do so by the device itself. In effect, the method would provide an external asynchronous input that would inform the processor that it should complete whatever instruction that is currently being executed and fetch a new routine that will service the requesting device. Once this servicing is complete however the processor would resume exactly where it left off.

This method is called **Interrupt**. It is easy to see that system throughput would drastically increase, and thus more tasks could be assumed by the microcomputer to further enhance its cost effectiveness.

The Programmable Interrupt Controller (PIC) functions as an overall manager in an Interrupt-Driven system environment. It accepts requests from the peripheral equipment, determines which of the incoming requests is of the highest importance (priority), ascertains whether the incoming request has a higher priority value than the level currently being serviced and issues an Interrupt to the CPU based on this determination.

Each peripheral device or structure usually has a special program or "routine" that is associated with its specific functional or operational requirements; this is referred to as a "service routine". The PIC, after issuing an Interrupt to the CPU, must somehow input information into the CPU that can "point" the Program Counter to the service routine associated with the requesting device. The PIC does this by providing the CPU with a 3-byte CALL instruction.

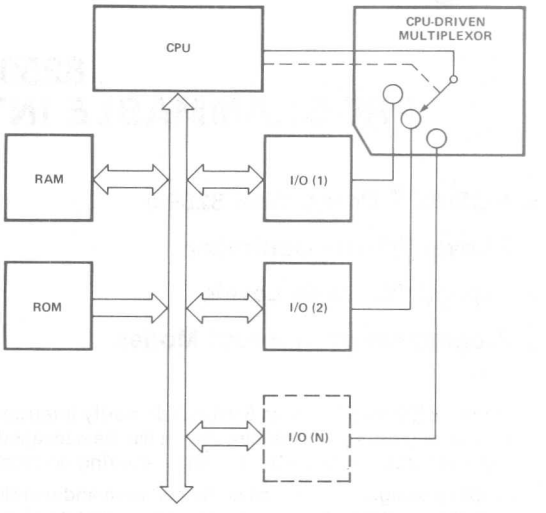


Figure 1. Polled Method

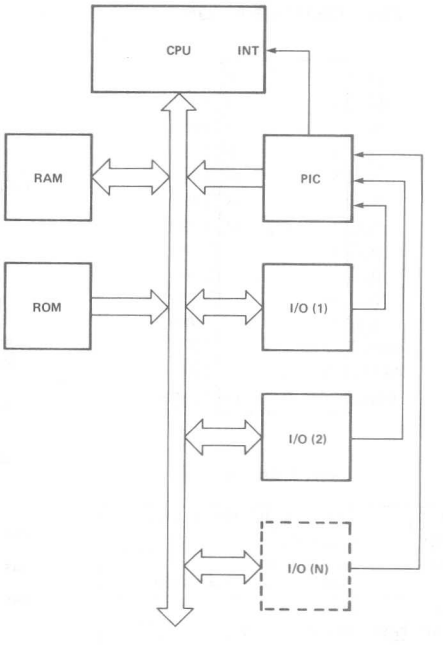


Figure 2. Interrupt Method

FUNCTIONAL DESCRIPTION

General

The 8259 is a device specifically designed for use in real time, interrupt driven, microcomputer systems. It manages eight levels or requests and has built-in features for expandability to other 8259s (up to 64 levels). It is programmed by the system's software as an I/O peripheral. A selection of priority modes is available to the programmer so that the requests are processed by the 8259 can be configured to match his system requirements. The priority modes can be changed or reconfigured dynamically at any time during the main program. This means that the complete interrupt structure can be defined as required, based on the total system environment.

Interrupt Request Register (IRR) and rice In-Service Register (ISR)

The interrupts at the IR input lines are handled by two registers in cascade, the Interrupt Request Register (IRR) and the In-Service Register (ISR). The IRR is used to store all the interrupt levels which are requesting service; and the ISR is used to store all the interrupt levels which are being serviced.

Priority Resolver

This logic block determines the priorities of the bits set in the IRR. The highest priority is selected and strobed into the corresponding bit of the ISR during $\overline{INTA}$ pulse.

INT (Interrupt)

This output goes directly to the CPU interrupt input. The V_{OH} level on this line is designed to be fully compatible with the 8080 input level.

$\overline{INTA}$ (Interrupt Acknowledge)

Three $\overline{INTA}$ pulses will cause the 8259 to release a 3-byte CALL instruction onto the Data Bus.

Interrupt Mask Register (IMR)

The IMR stores the bits of the interrupt lines to be masked. The IMR operates on the ISR. Masking of a higher priority input will not affect the interrupt request lines of lower priority.

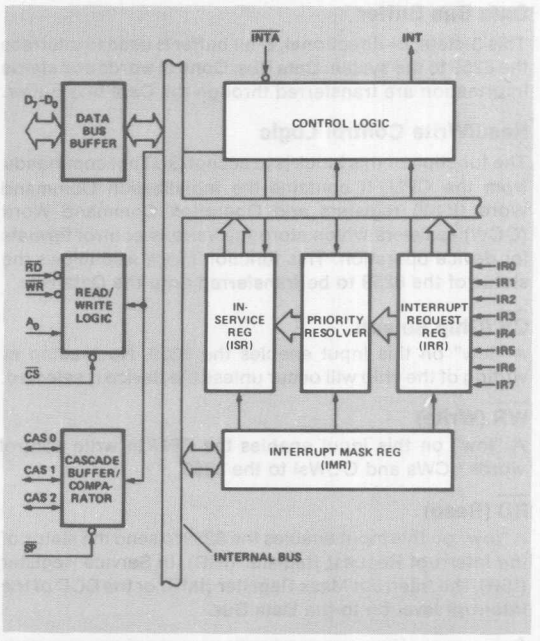


Figure 5. 8259 Block Diagram Showing Basic Interrupt Functions

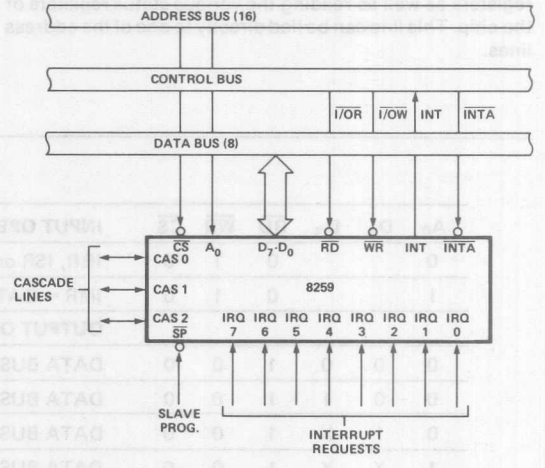


Figure 6. 8259 Interface to Standard System Bus

Data Bus Buffer

This 3-state, bi-directional, 8-bit buffer is used to interface the 8259 to the system Data Bus. Control words and status information are transferred through the Data Bus Buffer.

Read/Write Control Logic

The function of this block is to accept OUTput commands from the CPU. It contains the Initialization Command Word (ICW) registers and Operation Command Word (OCW) registers which store the various control formats for device operation. This function block also allows the status of the 8259 to be transferred onto the Data Bus.

$\overline{CS}$ (Chip Select)

A "low" on this input enables the 8259. No reading or writing of the chip will occur unless the device is selected.

$\overline{WR}$ (Write)

A "low" on this input enables the CPU to write control words (ICWs and OCWs) to the 8259.

$\overline{RD}$ (Read)

A "low" on this input enables the 8259 to send the status of the Interrupt Request Register (IRR), In Service Register (ISR), the Interrupt Mask Register (IMR) or the BCD of the Interrupt level on to the Data Bus.

A_0

This input signal is used in conjunction with $\overline{WR}$ and $\overline{RD}$ signals to write commands into the various command registers as well as reading the various status registers of the chip. This line can be tied directly to one of the address lines.

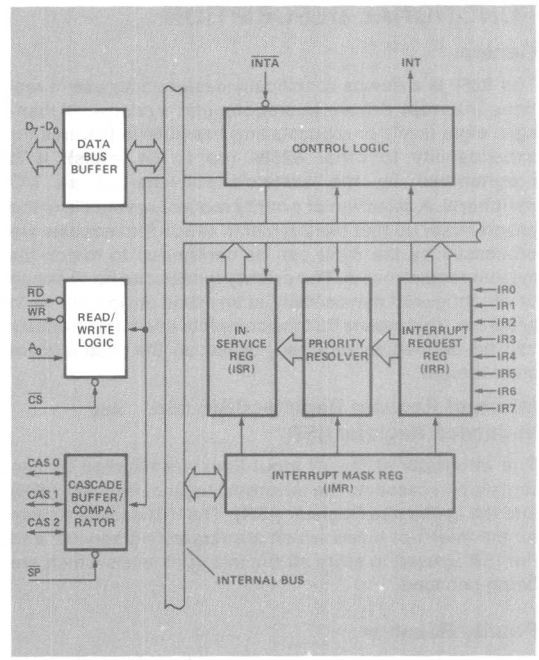


Figure 3. 8259 Block Diagram Showing Data Bus Buffer and Read/Write Logic Functions

A_0	D_4	D_3	$\overline{RD}$	$\overline{WR}$	$\overline{CS}$	INPUT OPERATION (READ)
0			0	1	0	IRR, ISR or Interrupting Level ⇒ DATA BUS (Note 1)
1			0	1	0	IMR ⇒ DATA BUS
						OUTPUT OPERATION (WRITE)
0	0	0	1	0	0	DATA BUS ⇒ OCW2
0	0	1	1	0	0	DATA BUS ⇒ OCW3
0	1	X	1	0	0	DATA BUS ⇒ ICW1
1	X	X	1	0	0	DATA BUS ⇒ OCW1, ICW2, ICW3 (Note 2)
						DISABLE FUNCTION
X	X	X	1	1	0	DATA BUS ⇒ 3-STATE
X	X	X	X	X	1	DATA BUS ⇒ 3-STATE

Note 1: Selection of IRR, ISR or Interrupting Level is based on the content of OCW3 written before the READ operation.
Note 2: On-chip sequencer logic queues these commands into proper sequence.

Figure 4. 8259 Basic Operation

SP (Slave Program)

More than one 8259 can be used in the system to expand the priority interrupt scheme up to 64 levels. In such case, one 8259 acts as the master, and the others act as slaves. A "high" on the $\overline{SP}$ pin designates the 8259 as the master, a "low" designates it as a slave.

The Cascade/Buffer/Comparator

This function block stores and compares the IDs of all 8259 used in the system. The associated three I/O pins (CAS0-2) are outputs when the 8259 is used as a master ($\overline{SP}$ = 1), and are inputs when the 8259 is used as a slave ($\overline{SP}$ = 0). As a master, the 8259 sends the ID of the interrupting slave device onto the CAS0-2 lines. The slave thus selected will send its preprogrammed subroutine addressed onto the Data Bus during next two consecutive $\overline{INTA}$ pulses. (See section "Cascading the 8259".)

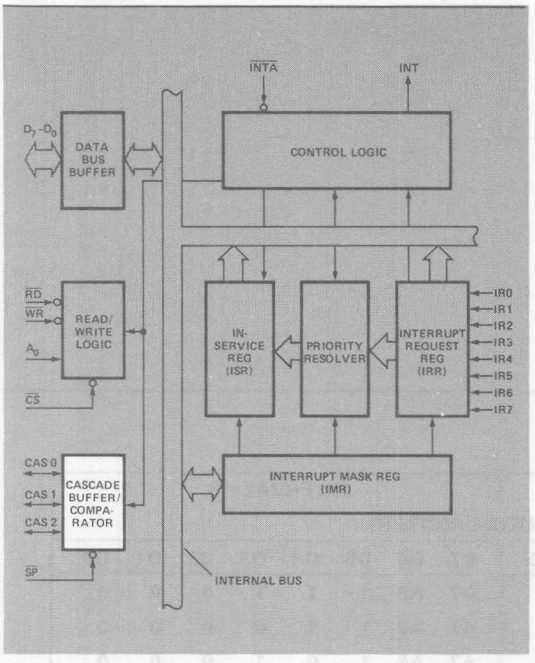


Figure 7. 8259 Block Diagram Showing Cascading Function

OPERATIONAL DESCRIPTION

General

The powerful features of the 8259 in a microcomputer system are its programmability and its utilization of the CALL instruction to jump into any address in the memory map. The normal sequence of events that the 8259 interacts with the CPU is as follows:

1. One or more of the INTERRUPT REQUEST lines (IR7-0) are raised high, setting the corresponding IRR bit(s).
2. The 8259 accepts these requests, resolves the priorities, and sends an INT to the CPU.
3. The CPU acknowledges the INT and responds with an $\overline{INTA}$ pulse.
4. Upon receiving an $\overline{INTA}$ from the CPU group, the highest priority ISR bit is set, and the corresponding IRR bit is reset. The 8259 will also release a CALL instruction code (11001101) onto the 8-bit Data Bus through its D7-0 pins.
5. This CALL instruction will initiate two more $\overline{INTA}$ pulses to be sent to the 8259 from the CPU group.
6. These two $\overline{INTA}$ pulses allow the 8259 to release its preprogrammed subroutine address onto the Data Bus. The lower 8-bit address is released at the first $\overline{INTA}$ pulse and the higher 8-bit address is released at the second $\overline{INTA}$ pulse.
7. This completes the 3-byte CALL instruction released by the 8259. ISR bit is not reset until the end of the subroutine when an EOI (End of interrupt) command is issued to the 8259.

Programming The 8259

The 8259 accepts two types of command words generated by the CPU:

1. Initialization Command Words (ICWs):
 Before normal operation can begin, each 8259 in the system must be brought to a starting point — by a sequence of 2 or 3 bytes timed by $\overline{WR}$ pulses. This sequence is described in Figure 1.
2. Operation Command Words (OCWs):
 These are the command words which command the 8259 to operate in various interrupt modes. These modes are:
 a. Fully nested mode
 b. Rotating priority mode
 c. Special mask mode
 d. Polled mode

The OCWs can be written into the 8259 at anytime after initialization.

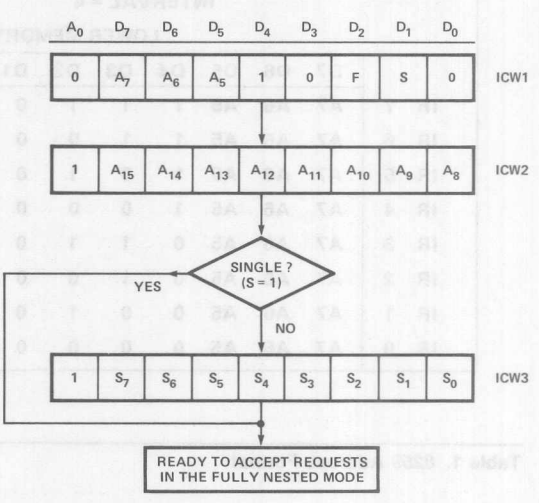


Figure 8. Initialization Sequence

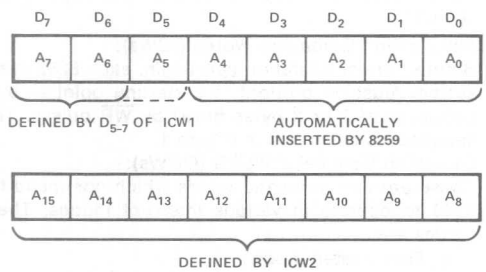
Initialization Command Words 1 and 2 (ICW1 and ICW2)

Whenever a command is issued with A0 = 0 and D4 = 1, this is interpreted as Initialization Command Word 1 (ICW1), and initiates the initialization sequence. During this sequence, the following occur automatically:

a. The edge sense circuit is reset, which means that following initialization, an interrupt request (IR) input must make a low to high transition to generate an interrupt.
b. The interrupt Mask Register is cleared.
c. IR 7 input is assigned priority 7.
d. Special Mask Mode Flip-flop and status Read Flip-flop are reset.

The 8 requesting devices have 8 addresses equally spaced in memory. The addresses can be programmed at intervals of 4 or 8 bytes; the 8 routines thus occupying a page of 32 or 64 bytes respectively in memory.

The address format is:



A0-4 are automatically inserted by the 8259, while A15-6 are programmed by ICW1 and ICW2. When interval = 8, A5 is fixed by the 8259. If interval = 4, A5 is programmed in ICW1. Thus, the interrupt service routines can be located anywhere in the memory space. The 8 byte interval will maintain compatibility with current 8080 RESTART instruction software, while the 4 byte interval is best for compact jump table.

The address format inserted by the 8259 is described in Table 1.

The bits F and S are defined by ICW1 as follows:

F: Call address interval. F = 1, then interval = 4; F = 0, then interval = 8.

S: Single. S = 1 means that this is the only 8259 in the system. It avoids the necesity of programming ICW3.

	INTERVAL = 4								INTERVAL = 8							
	LOWER MEMORY ROUTINE ADDRESS															
	D7	D6	D5	D4	D3	D2	D1	D0	D7	D6	D5	D4	D3	D2	D1	D0
IR 7	A7	A6	A5	1	1	1	0	0	A7	A6	1	1	1	0	0	0
IR 6	A7	A6	A5	1	1	0	0	0	A7	A6	1	1	0	0	0	0
IR 5	A7	A6	A5	1	0	1	0	0	A7	A6	1	0	1	0	0	0
IR 4	A7	A6	A5	1	0	0	0	0	A7	A6	1	0	0	0	0	0
IR 3	A7	A6	A5	0	1	1	0	0	A7	A6	0	1	1	0	0	0
IR 2	A7	A6	A5	0	1	0	0	0	A7	A6	0	1	0	0	0	0
IR 1	A7	A6	A5	0	0	1	0	0	A7	A6	0	0	1	0	0	0
IR 0	A7	A6	A5	0	0	0	0	0	A7	A6	0	0	0	0	0	0

Table 1. 8259 Address Format

Example of Interrupt Acknowledge Sequence

Assume the 8259 is programmed with F = 1 (CALL address interval = 4), and IR5 is the interrupting level. The 3 byte sequence released by the 8259 timed by the $\overline{INTA}$ pulses is as follows:

	D7	D6	D5	D4	D3	D2	D1	D0	
1st $\overline{INTA}$	1	1	0	0	1	1	0	1	CALL CODE
2nd $\overline{INTA}$	A7	A6	A5	1	0	1	0	0	LOWER ROUTINE ADDRESS
3rd $\overline{INTA}$	A15	A14	A13	A12	A11	A10	A9	A8	HIGHER ROUTINE ADDRESS

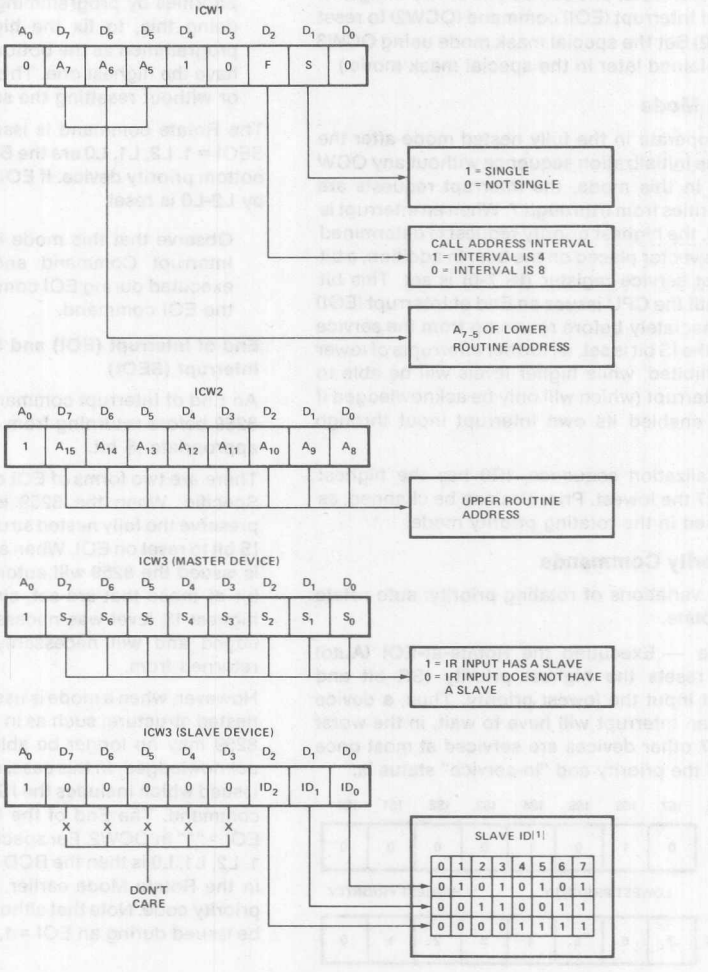
Initialization Command Word 3 (ICW3)

This will load the 8-bit slave register. The functions of this register are as follows:

a. If the 8259 is the master, a "1" is set for each slave in the system. The master then will release byte 1 of the CALL sequence and will enable the corresponding slave to release bytes 2 and 3, through the cascade lines.

b. If the 8259 is a slave, bits 2 - 0 identify the slave. The slave compares its CAS0-2 inputs (sent by the master) with these bits. If they are equal, bytes 2 and 3 of the CALL sequence are released.

If bit S is set in ICW1, there is no need to program ICW3.



NOTE 1: SLAVE ID IS EQUAL TO THE CORRESPONDING MASTER IR INPUT.

Figure 9. Initialization Command Word Format

Operation Command Words (OCWs)

After the Initialization Command Words (ICWs) are programmed into the 8259, the chip is ready to accept interrupt requests at its input lines. However, during the 8259 operation, a selection of algorithms can command the 8259 to operate in various modes through the Operation Command Words (OCWs). These various modes and their associated OCWs are described below.

Interrupt Masks

Each Interrupt Request input can be masked individually by the Interrupt Masked Register (IMR) programmed through OCW1.

The IMR operates on the In-Service Register. Note that if an interrupt is already acknowledged by the 8259 (an $\overline{INTA}$ pulse has occurred), then the Interrupting level, although masked, will inhibit the lower priorities. To enable these lower priority interrupts, one can do one of two things: (1) Write an End of Interrupt (EOI) command (OCW2) to reset the ISR bit or (2) Set the special mask mode using OCW3 (as will be explained later in the special mask mode.)

Fully Nested Mode

The 8259 will operate in the fully nested mode after the execution of the initialization sequence without any OCW being written. In this mode, the interrupt requests are ordered in priorities from 0 through 7. When an interrupt is acknowledged, the highest priority request is determined and its address vector placed on the bus. In addition, a bit of the Interrupt service register (IS 7-0) is set. This bit remains set until the CPU issues an End of Interrupt (EOI) command immediately before returning from the service routine. While the IS bit is set, all further interrupts of lower priority are inhibited, while higher levels will be able to generate an interrupt (which will only be acknowledged if the CPU has enabled its own interrupt input through software).

After the Initialization sequence, IR0 has the highest priority and IR7 the lowest. Priorities can be changed, as will be explained in the rotating priority mode.

Rotating Priority Commands

There are two variations of rotating priority: auto rotate and specific rotate.

1. Auto Rotate — Executing the Rotate-at-EOI (Auto) command, resets the highest priority ISR bit and assigns that input the lowest priority. Thus, a device requesting an interrupt will have to wait, in the worst case, until 7 other devices are serviced at most once each, i.e., if the priority and "in-service" status is:

BEFORE ROTATE	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
"IS" STATUS	0	1	0	1	0	0	0	0
	LOWEST PRIORITY						HIGHEST PRIORITY	
PRIORITY STATUS	7	6	5	4	3	2	1	0

AFTER ROTATE	IS7	IS6	IS5	IS4	IS3	IS2	IS1	IS0
"IS" STATUS	0	1	0	0	0	0	0	0
	LOWEST PRIORITY						HIGHEST PRIORITY	
PRIORITY STATUS	4	3	2	1	0	7	6	5

In this example, the In-Service FF corresponding to line 4 (the highest priority FF set) was reset and line 4 became the lowest priority, while all the other priorities rotated correspondingly.

The Rotate command is issued in OCW2, where: R = 1, EOI = 1, SEOI = 0.

2. Specific Rotate — The programmer can change priorities by programming the bottom priority, and by doing this, to fix the highest priority: i.e., if IR5 is programmed as the bottom priority device, the IR6 will have the highest one. This command can be used with or without resetting the selected ISR bit.

The Rotate command is issued in OCW2 where: R = 1, SEOI = 1. L2, L1, L0 are the BCD priority level codes of the bottom priority device. If EOI = 1 also, the ISR bit selected by L2-L0 is reset.

Observe that this mode is independent of the End of Interrupt Command and priority changes can be executed during EOI command or independently from the EOI command.

End of Interrupt (EOI) and Specific End of Interrupt (SEOI)

An End of Interrupt command word must be issued to the 8259 before returning from a service routine, to reset the appropriate IS bit.

There are two forms of EOI command: Specific and non-Specific. When the 8259 is operated in modes which preserve the fully nested structure, it can determine which IS bit to reset on EOI. When a non-Specific EOI command is issued the 8259 will automatically reset the highest IS bit of those that are set, since in the nested mode, the highest IS level was necessarily the last level acknowledged and will necessarily be the next routine level returned from.

However, when a mode is used which may disturb the fully nested structure, such as in the rotating priority case, the 8259 may no longer be able to determine the last level acknowledged. In this case, a specific EOI (SEOI) must be issued which includes the IS level to be reset as part of the command. The End of the Interrupt is issued whenever EOI = "1" in OCW2. For specific EOI, SEOI = "1", and EOI = 1. L2, L1, L0 is then the BCD level to be reset. As explained in the Rotate Mode earlier, this can also be the bottom priority code. Note that although the Rotate command can be issued during an EOI = 1, it is not necessarily tied to it.

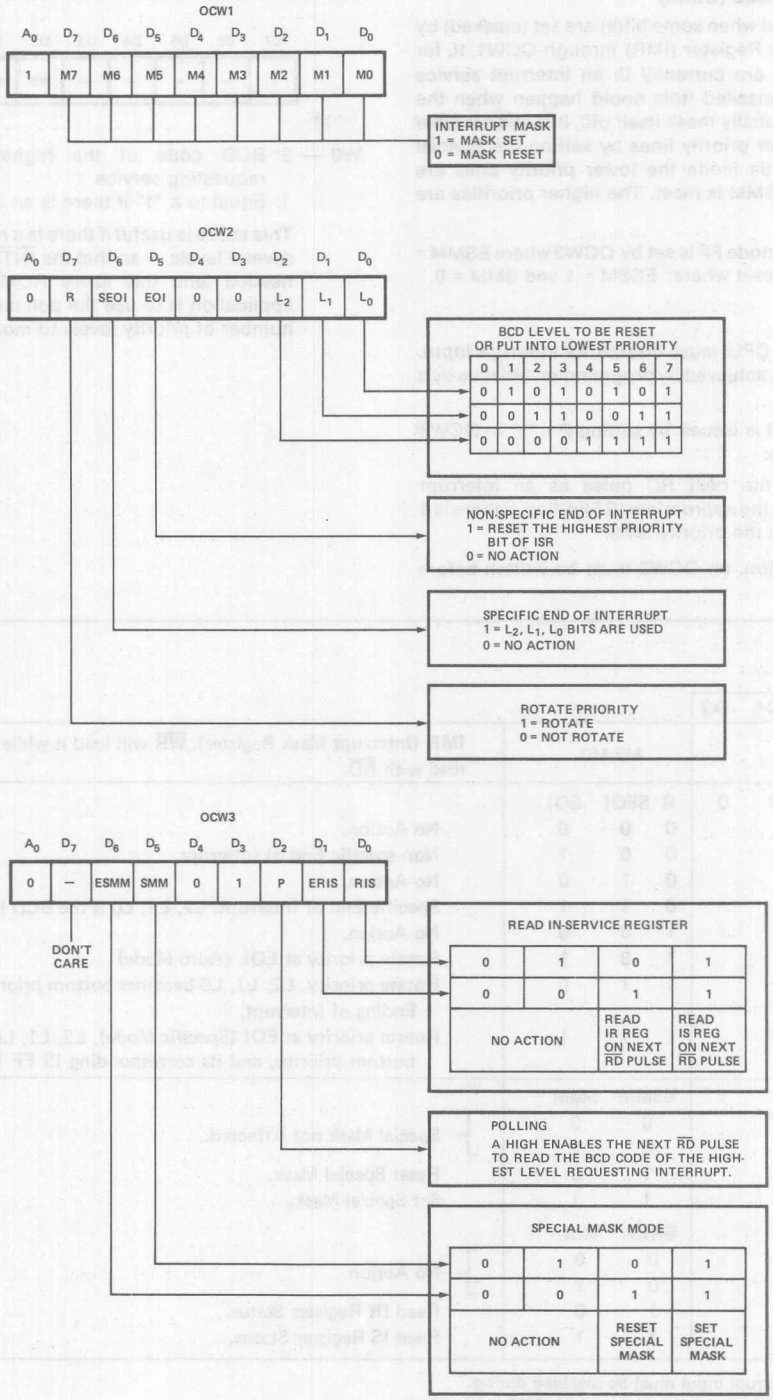


Figure 10. Operation Command Word Format

Special Mask Mode (SMM)

This mode is useful when some bit(s) are set (masked) by the Interrupt Mask Register (IMR) through OCW1. If, for some reason, we are currently in an interrupt service routine which is masked (this could happen when the subroutine intentionally mask itself off), it is still possible to enable the lower priority lines by setting the Special Mask mode. In this mode the lower priority lines are enabled until the SMM is reset. The higher priorities are not affected.

The special mask mode FF is set by OCW3 where ESMM = 1, SMM = 1, and reset where: ESSM = 1 and SMM = 0.

Polled Mode

In this mode, the CPU must disable its interrupt input. Service to device is achieved by programmer initiative by a Poll command.

The poll command is issued by setting P = "1" in OCW3 during a $\overline{WR}$ pulse.

The 8259 treats the next $\overline{RD}$ pulse as an interrupt acknowledge, sets the appropriate IS Flip-flop, if there is a request, and reads the priority level.

For polling operation, an OCW3 must be written before every read.

The word enabled onto the data bus during $\overline{RD}$ is:

D7	D6	D5	D4	D3	D2	D1	D0
I	–	–	–	–	W2	W1	W0

W0 — 2: BCD code of the highest priority level requesting service.

I: Equal to a "1" if there is an interrupt.

This mode is useful if there is a routine common to several levels — so that the $\overline{INTA}$ sequence is not needed (and this saves ROM space). Another application is to use the poll mode to expand the number of priority levels to more than 64.

	A0	D4	D3		
OCW1	1			M7-M0	IMR (Interrupt Mask Register). $\overline{WR}$ will load it while status can be read with $\overline{RD}$.
OCW2	0	0	0	R SEOI EOI	
				0 0 0	No Action.
				0 0 1	Non-specific End of Interrupt.
				0 1 0	No Action.
				0 1 1	Specific End of Interrupt. L2, L1, L0 is the BCD level to be reset.
				1 0 0	No Action.
				1 0 1	Rotate priority at EOI. (Auto Mode)
				1 1 0	Rotate priority, L2, L1, L0 becomes bottom priority without Ending of Interrupt.
				1 1 1	Rotate priority at EOI (Specific Mode), L2, L1, L0 becomes bottom priority, and its corresponding IS FF is reset.
OCW3	0	0	1	ESMM SMM	
				0 0	Special Mask not Affected.
				0 1	
				1 0	Reset Special Mask.
				1 1	Set Special Mask.
				ERIS RIS	
				0 0	No Action.
				0 1	
				1 0	Read IR Register Status.
				1 1	Read IS Register Status.

Note: The CPU interrupt input must be disabled during:
1. Initialization sequence for all the 8259 in the system.
2. Any control command execution.

Figure 11. Summary of Operation Word Programming

Reading 8259 Status

The input status of several internal registers can be read to update the user information on the system. The following registers can be read by issuing a suitable OCW3 and reading with $\overline{RD}$.

Interrupt Requests Register (IRR): 8-bit register which contains the levels requesting an interrupt to be acknowledged. The highest request level is reset from the IRR when an interrupt is acknowledged. (Not affected by IMR).

In Service Register (ISR): 8-bit register which contains the priority levels that are being serviced. The ISR is updated when an End of Interrupt command is issued.

Interrupt Mask Register: 8-bit register which contains the interrupt request lines which are masked.

The IRR can be read when prior to the $\overline{RD}$ pulse, an $\overline{WR}$ pulse is issued with OCW3, and ERIS = 1, RIS = 0.

The ISR can be read in a similar mode, when ERIS = 1, RIS = 1.

There is no need to write an OCW3 before every status read operation as long as the status read corresponds with the previous one, i.e. the 8259 "remembers" whether the IRR or ISR has been previously selected by the OCW3.

For reading the IMR, a $\overline{WR}$ pulse is not necessary to preceed the $\overline{RD}$. The output data bus will contain the IMR whenever $\overline{RD}$ is active and A0 = 1.

Polling overrides status read when P = 1, ERIS = 1 in OCW3.

Cascading

The 8259 can be easily interconnected in a system of one master with up to eight slaves to handle up to 64 priority levels.

A typical system is shown in Figure 12. The master controls, through the 3 line cascade bus, which one of the slaves will release the corresponding address.

As shown in Figure 12, the slaves interrupt outputs are connected to the master interrupt request inputs. When a slave request line is activated and later acknowledged, the master releases the 8080 CALL code during the first $\overline{INTA}$ pulse. From the trailing edge of this first $\overline{INTA}$ pulse until the trailing edge of the third pulse, the CAS lines will contain the slave address code. Thus, the corresponding slave is enabled to release the two-byte service routine address during the second and third $\overline{INTA}$ pulses.

Note that since the CAS lines default to 000, no slave should be connected with IR0 on the master unless all other master request inputs (IR1-IR7) are connected to slaves. Otherwise, the slave on IR0 will attempt to drive the data bus in conflict with a non-slave interrupt request on the master.

It is obvious that each 8259 in the system must follow a separate initialization sequence and can be programmed to work in a different mode. An EOI command must be issued twice: once for the master and once for the corresponding slave. An address decoder is required to activate the Chip Select ($\overline{CS}$) input of each 8259. The slave program pin ($\overline{SP}$) must be at a "low" level for a slave (and then the cascade lines are inputs) and at a "high" level for a master (and then the cascade lines are outputs).

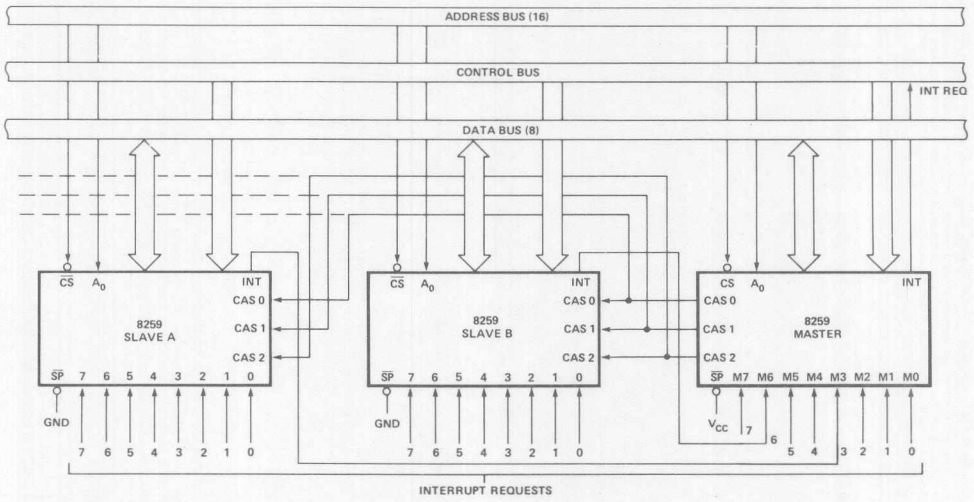


Figure 12. Cascading the 8259

INST. NO.		A0	D7	D6	D5	D4	D3	D2	D1	D0	OPERATION DESCRIPTION
1	ICW1 A	0	A7	A6	A5	1	0	1	1	0	Byte 1 initialization, format = 4, single.
2	ICW1 B	0	A7	A6	A5	1	0	1	0	0	Byte 1 initialization, format = 4, not single.
3	ICW1 C	0	A7	A6	A5	1	0	0	1	0	Byte 1 initialization, format = 8, single.
4	ICW1 D	0	A7	A6	A5	1	0	0	0	0	Byte 1 initialization, format = 8, not single.
5	ICW2	1	A15	A14	A13	A12	A11	A10	A9	A8	Byte 2 initialization (Address No. 2)
6	ICW3 M	1	S7	S6	S5	S4	S3	S2	S1	S0	Byte 3 initialization — master.
7	ICW3 S	1	0	0	0	0	0	S2	S1	S0	Byte 3 initialization — slave.
8	OCW1	1	M7	M6	M5	M4	M3	M2	M1	M0	Load mask reg, read mask reg.
9	OCW2 E	0	0	0	1	0	0	0	0	0	Non specific EOI.
10	OCW2 SE	0	0	1	1	0	0	L2	L1	L0	Specific EOI. L2, L1, L0 code of IS FF to be reset.
11	OCW2 RE	0	1	0	1	0	0	0	0	0	Rotate at EOI (Auto Mode).
12	OCW2 RSE	0	1	1	1	0	0	L2	L1	L0	Rotate at EOI (Specific Mode). L2, L1, L0, code of line to be reset and selected as bottom priority.
13	OCW2 RS	0	1	1	0	0	0	L2	L1	L0	L2, L1, L0 code of bottom priority line.
14	OCW3 P	0	—	0	0	0	1	1	0	0	Poll mode.
15	OCW3 RIS	0	—	0	0	0	1	0	1	1	Read IS register.
16	OCW3 RR	0	—	0	0	0	1	0	1	0	Read requests register.
17	OCW3 SM	0	—	1	1	0	1	0	0	0	Set special mask mode.
18	OCW3 RSM	0	—	1	0	0	1	0	0	0	Reset special mask mode.

Notes:
1. In the master mode $\overline{SP}$ pin = 1, in slave mode $\overline{SP}$ = 0.
2. (—) = do not care.

Figure 13. 8259 Instruction Set

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias 0°C to 70°C
Storage Temperature −65°C to +150°C
Voltage On Any Pin
With Respect to Ground −0.5 V to +7 V
Power Dissipation 1 Watt

*COMMENT:
Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

D.C. CHARACTERISTICS

(T_A = 0°C to 70°C; V_{CC} = 5V ±5%)

SYMBOL	PARAMETER	MIN.	MAX.	UNITS	TEST CONDITIONS
V_{IL}	Input Low Voltage	-.5	.8	V	
V_{IH}	Input High Voltage	2.0	V_{CC}+.5V	V	
V_{OL}	Output Low Voltage		.45	V	I_{OL} = 2 mA
V_{OH}	Output High Voltage	2.4		V	I_{OH} = -400 μA
V_{OH-INT}	Interrupt Output High Voltage	2.4		V	I_{OH} = -400 μA
		3.5		V	I_{OH} = -50 μA
$I_{IL(IR_{0-7})}$	Input Leakage Current		-300	μA	V_{IN} = 0V
	for IR_{0-7}		10	μA	V_{IN} = V_{CC}
I_{IL}	Input Leakage Current for Other Inputs		10	μA	V_{IN} = V_{CC} to 0V
I_{OFL}	Output Float Leakage		±10	μA	V_{OUT} = 0.45V to V_{CC}
I_{CC}	V_{CC} Supply Current		100	mA	

CAPACITANCE

T_A = 25°C; V_{CC} = GND = 0V

SYMBOL	PARAMETER	MIN.	TYP.	MAX.	UNIT	TEST CONDITIONS
C_{IN}	Input Capacitance			10	pF	fc = 1 MHz
$C_{I/O}$	I/O Capacitance			20	pF	Unmeasured pins returned to V_{SS}

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

A.C. CHARACTERISTICS

(T_A = 0°C to 70°C; V_{CC} = +5V ±5%, GND = 0V)

Bus Parameters

Read:

SYMBOL	PARAMETER	8259 MIN.	8259 MAX.	8259-5 MIN.	8259-5 MAX.	UNIT
t_{AR}	$\overline{CS}/A_0$ Stable Before $\overline{RD}$ or $\overline{INTA}$	50		50		ns
t_{RA}	$\overline{CS}/A_0$ Stable After $\overline{RD}$ or $\overline{INTA}$	5		30		ns
t_{RR}	$\overline{RD}$ Pulse Width	420		300		ns
t_{RD}	Data Valid From $\overline{RD}/\overline{INTA}$[1]		300		200	ns
t_{DF}	Data Float After $\overline{RD}/\overline{INTA}$	20	200	20	100	ns

Write:

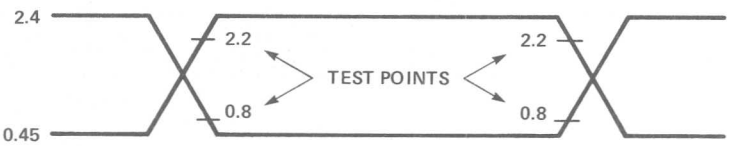
SYMBOL	PARAMETER	8259 MIN.	8259 MAX.	8259-5 MIN.	8259-5 MAX.	UNIT
t_{AW}	A_0 Stable Before $\overline{WR}$	50		50		ns
t_{WA}	A_0 Stable After $\overline{WR}$	20		30		ns
t_{WW}	$\overline{WR}$ Pulse Width	400		300		ns
t_{DW}	Data Valid to $\overline{WR}$ (T.E.)	300		250		ns
t_{WD}	Data Valid After $\overline{WR}$	40		30		ns

Other Timings:

SYMBOL	PARAMETER	8259 MIN.	8259 MAX.	8259-5 MIN.	8259-5 MAX.	UNIT
t_{IW}	Width of Interrupt Request Pulse	100		100		ns
t_{INT}	INT ↑ After IR ↑	400		350		ns
t_{IC}	Cascade Line Stable After $\overline{INTA}$ ↑	400		400		ns

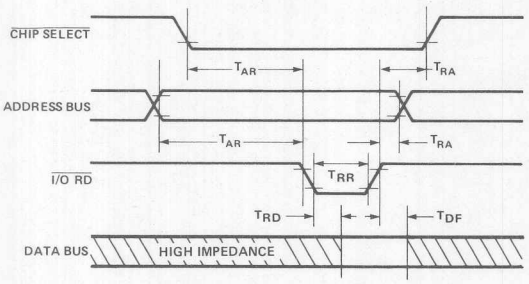
Note 1: 8259: C_L = 100pF, 8259-5: C_L = 150pF.

Input Waveforms for A.C. Tests

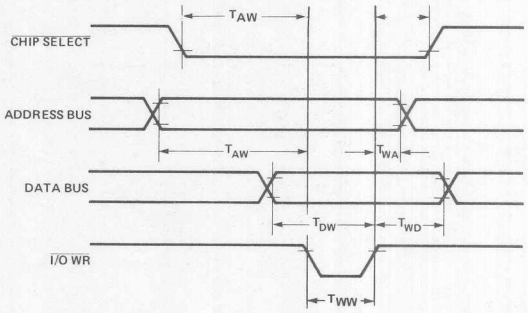


WAVEFORMS

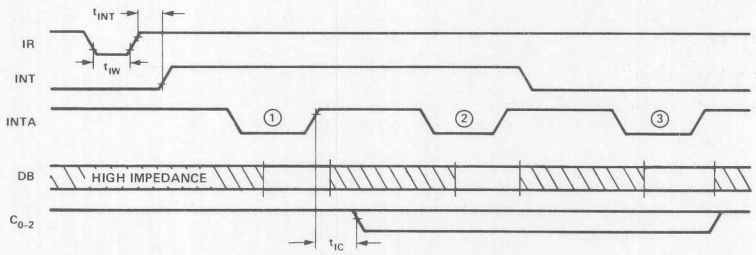
Read Timing



Write Timing

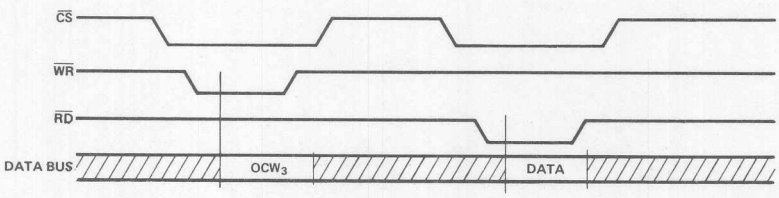


Other Timing



Note: Interrupt Request must remain "HIGH" (at least) until leading edge of first INTA.

Read Status/Poll Mode



PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

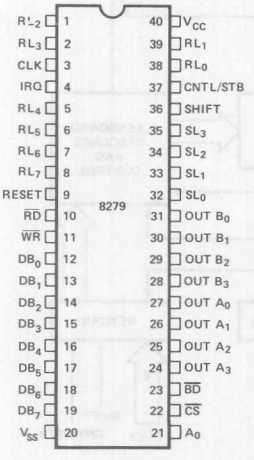
8279/8279-5
PROGRAMMABLE KEYBOARD/DISPLAY INTERFACE

- MCS-85™ Compatible 8279-5
- Simultaneous Keyboard Display Operations
- Scanned Keyboard Mode
- Scanned Sensor Mode
- Strobed Input Entry Mode
- 8-Character Keyboard FIFO
- 2-Key Lockout or N-Key Rollover with Contact Debounce
- Dual 8- or 16-Numerical Display
- Single 16-Character Display
- Right or Left Entry 16-Byte Display RAM
- Mode Programmable from CPU
- Programmable Scan Timing
- Interrupt Output on Key Entry

The Intel® 8279 is a general purpose programmable keyboard and display I/O interface device designed for use with Intel® microprocessors. The keyboard portion can provide a scanned interface to a 64-contact key matrix. The keyboard portion will also interface to an array of sensors or a strobed interface keyboard, such as the hall effect and ferrite variety. Key depressions can be 2-key lockout or N-key rollover. Keyboard entries are debounced and strobed in an 8-character FIFO. If more than 8 characters are entered, overrun status is set. Key entries set the interrupt output line to the CPU.

The display portion provides a scanned display interface for LED, incandescent, and other popular display technologies. Both numeric and alphanumeric segment displays may be used as well as simple indicators. The 8279 has 16X8 display RAM which can be organized into dual 16X4. The RAM can be loaded or interrogated by the CPU. Both right entry, calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.

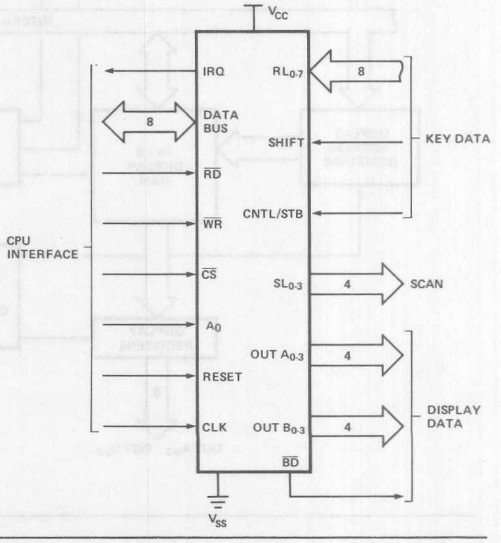
PIN CONFIGURATION



PIN NAMES

DB_{0-7}	I/O	DATA BUS (BI-DIRECTIONAL)
CLK	I	CLOCK INPUT
RESET	I	RESET INPUT
$\overline{CS}$	I	CHIP SELECT
$\overline{RD}$	I	READ INPUT
$\overline{WR}$	I	WRITE INPUT
A_0	I	BUFFER ADDRESS
IRQ	O	INTERRUPT REQUEST OUTPUT
SL_{0-3}	O	SCAN LINES
RL_{0-7}	I	RETURN LINES
SHIFT	I	SHIFT INPUT
CNTL/STB	I	CONTROL/STROBE INPUT
$OUT\ A_{0-3}$	O	DISPLAY (A) OUTPUTS
$OUT\ B_{0-3}$	O	DISPLAY (B) OUTPUTS
$\overline{BD}$	O	BLANK DISPLAY OUTPUT

LOGIC SYMBOL



© Intel Corporation, 1978

February, 1978

FUNCTIONAL DESCRIPTION

Since data input and display are an integral part of many microprocessor designs, the system designer needs an interface that can control these functions without placing a large load on the CPU. The 8279 provides this function for 8-bit microprocessors.

The 8279 has two sections: keyboard and display. The keyboard section can interface to regular typewriter style keyboards or random toggle or thumb switches. The display section drives alphanumeric displays or a bank of indicator lights. Thus the CPU is relieved from scanning the keyboard or refreshing the display.

The 8279 is designed to directly connect to the microprocessor bus. The CPU can program all operating modes for the 8279. These modes include:

Input Modes

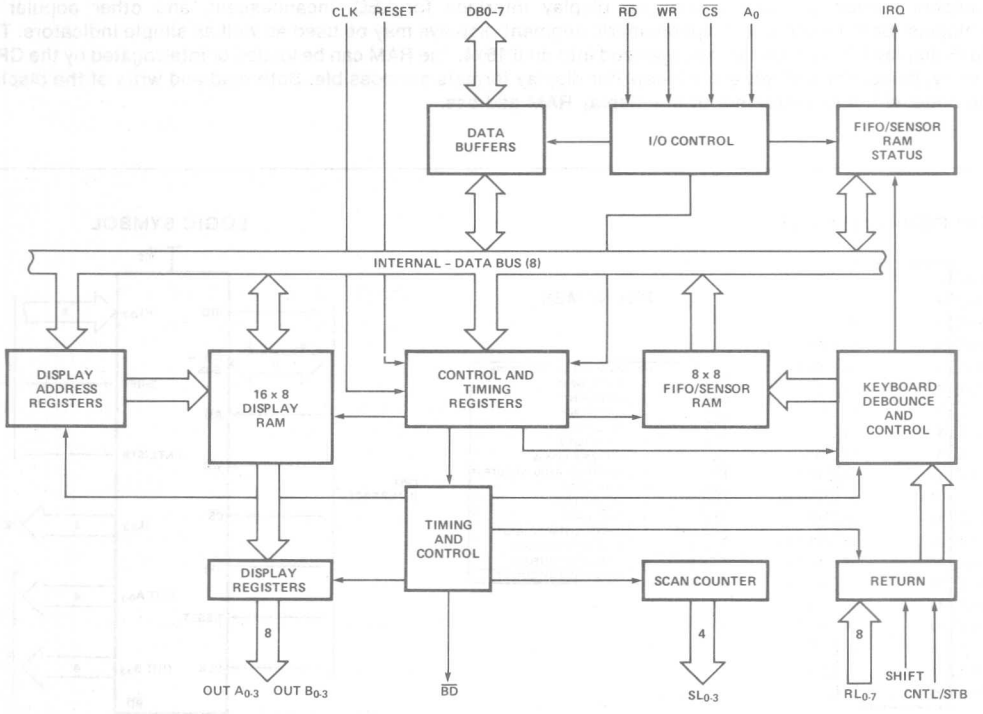
- Scanned Keyboard — with encoded (8 x 8 x 4 key keyboard) or decoded (4 x 8 x 4 key keyboard) scan lines. A key depression generates a 6-bit encoding of key position. Position and shift and control status are stored in the FIFO. Keys are automatically debounced with 2-key lockout or N-key rollover.
- Scanned Sensor Matrix — with encoded (8 x 8 matrix switches) or decoded (4 x 8 matrix switches) scan lines. Key status (open or closed) stored in RAM addressable by CPU.
- Strobed Input — Data on return lines during control line strobe is transferred to FIFO.

Output Modes

- 8 or 16 character multiplexed displays that can be organized as dual 4-bit or single 8-bit.
- Right entry or left entry display formats.

Other features of the 8279 include:

- Mode programming from the CPU.
- Programmable clock to match the 8279 scan times to the CPU cycle time.
- Interrupt output to signal CPU when there is keyboard or sensor data available.
- An 8 byte FIFO to store keyboard information.
- 16 byte internal Display RAM for display refresh. This RAM can also be read by the CPU.



HARDWARE DESCRIPTION

The 8279 is packaged in a 40 pin DIP. The following is a functional description of each pin.

No. Of Pins	Designation	Function
8	DB_0-DB_7	Bi-directional data bus. All data and commands between the CPU and the 8279 are transmitted on these lines.
1	CLK	Clock from system used to generate internal timing.
1	RESET	A high signal on this pin resets the 8279.
1	$\overline{CS}$	Chip Select. A low on this pin enables the interface functions to receive or transmit.
1	A_0	Buffer Address. A high on this line indicates the signals in or out are interpreted as a command or status. A low indicates that they are data.
2	$\overline{RD}$, $\overline{WR}$	Input/Output read and write. These signals enable the data buffers to either send data to the external bus or receive it from the external bus.
1	IRQ	Interrupt Request. In a keyboard mode, the interrupt line is high when there is data in the FIFO/Sensor RAM. The interrupt line goes low with each FIFO/Sensor RAM read and returns high if there is still information in the RAM. In a sensor mode, the interrupt line goes high whenever a change in a sensor is detected.
2	V_{SS}, V_{CC}	Ground and power supply pins.
4	SL_0-SL_3	Scan Lines which are used to scan the key switch or sensor matrix and the display digits. These lines can be either encoded (1 of 16) or decoded (1 of 4).
8	RL_0-RL_7	Return line inputs which are connected to the scan lines through the keys or sensor switches. They have active internal pullups to keep them high until a switch closure pulls one low. They also serve as an 8-bit input in the Strobed Input mode.
1	SHIFT	The shift input status is stored along with the key position on key closure in the Scanned Keyboard modes. It has an active internal pullup to keep it high until a switch closure pulls it low.
1	CNTL/STB	For keyboard modes this line is used as a control input and stored like status on a key closure. The line is also the strobe line that enters the data into the FIFO in the Strobed Input mode. (Rising Edge). It has an active internal pullup to keep it high until a switch closure pulls it low.
4 4	OUT A_0-OUT A_3 OUT B_0-OUT B_3	These two ports are the outputs for the 16 x 4 display refresh registers. The data from these outputs is synchronized to the scan lines (SL_0-SL_3) for multiplexed digit displays. The two 4 bit ports may be blanked independently. These two ports may also be considered as one 8 bit port.
1	$\overline{BD}$	Blank Display. This output is used to blank the display during digit switching or by a display blanking command.

PRINCIPLES OF OPERATION

The following is a description of the major elements of the 8279 Programmable Keyboard/Display interface device. Refer to the block diagram in Figure 1.

I/O Control and Data Buffers

The I/O control section uses the $\overline{CS}$, A_0, $\overline{RD}$ and $\overline{WR}$ lines to control data flow to and from the various internal registers and buffers. All data flow to and from the 8279 is enabled by $\overline{CS}$. The character of the information, given or desired by the CPU, is identified by A_0. A logic one means the information is a command or status. A logic zero means the information is data. $\overline{RD}$ and $\overline{WR}$ determine the direction of data flow through the Data Buffers. The Data Buffers are bi-directional buffers that connect the internal bus to the external bus. When the chip is not selected ($\overline{CS}$ = 1), the devices are in a high impedance state. The drivers input during $\overline{WR} \bullet \overline{CS}$ and output during $\overline{RD} \bullet \overline{CS}$.

Control and Timing Registers and Timing Control

These registers store the keyboard and display modes and other operating conditions programmed by the CPU. The modes are programmed by presenting the proper command on the data lines with A_0 = 1 and then sending a $\overline{WR}$. The command is latched on the rising edge of $\overline{WR}$.

The command is then decoded and the appropriate function is set. The timing control contains the basic timing counter chain. The first counter is a ÷ N prescaler that can be programmed to match the CPU cycle time to the internal timing. The prescaler is software programmed to a value between 2 and 31. A value which yields an internal frequency of 100 kHz gives a 5.1 ms keyboard scan time and a 10.3 ms debounce time. The other counters divide down the basic internal frequency to provide the proper key scan, row scan, keyboard matrix scan, and display scan times.

Scan Counter

The scan counter has two modes. In the encoded mode, the counter provides a binary count that must be externally decoded to provide the scan lines for the keyboard and display. In the decoded mode, the scan counter decodes the least significant 2 bits and provides a decoded 1 of 4 scan. Note than when the keyboard is in decoded scan, so is the display. This means that only the first 4 characters in the Display RAM are displayed.

In the encoded mode, the scan lines are active high outputs. In the decoded mode, the scan lines are active low outputs.

Return Buffers and Keyboard Debounce and Control

The 8 return lines are buffered and latched by the Return Buffers. In the keyboard mode, these lines are scanned, looking for key closures in that row. If the debounce circuit detects a closed switch, it waits about 10 msec to check if the switch remains closed. If it does, the address of the switch in the matrix plus the status of SHIFT and CONTROL are transferred to the FIFO. In the scanned Sensor Matrix modes, the contents of the return lines is directly transferred to the corresponding row of the Sensor RAM (FIFO) each key scan time. In Strobed Input mode, the contents of the return lines are transferred to the FIFO on the rising edge of the CNTL/STB line pulse.

FIFO/Sensor RAM and Status

This block is a dual function 8 x 8 RAM. In Keyboard or Strobed Input modes, it is a FIFO. Each new entry is written into successive RAM positions and each is then read in order of entry. FIFO status keeps track of the number of characters in the FIFO and whether it is full or empty. Too many reads or writes will be recognized as an error. The status can be read by an $\overline{RD}$ with $\overline{CS}$ low and A_0 high. The status logic also provides an IRQ signal when the FIFO is not empty. In Scanned Sensor Matrix mode, the memory is a Sensor RAM. Each row of the Sensor RAM is loaded with the status of the corresponding row of sensor in the sensor matrix. In this mode, IRQ is high if a change in a sensor is detected.

Display Address Registers and Display RAM

The Display Address Registers hold the address of the word currently being written or read by the CPU and the two 4-bit nibbles being displayed. The read/write addresses are programmed by CPU command. They also can be set to auto increment after each read or write. The Display RAM can be directly read by the CPU after the correct mode and address is set. The addresses for the A and B nibbles are automatically updated by the 8279 to match data entry by the CPU. The A and B nibbles can be entered independently or as one word, according to the mode that is set by the CPU. Data entry to the display can be set to either left or right entry. See Interface Considerations for details.

SOFTWARE OPERATION

8279 commands

The following commands program the 8279 operating modes. The commands are sent on the Data Bus with $\overline{CS}$ low and A_0 high and are loaded to the 8279 on the rising edge of $\overline{WR}$.

Keyboard/Display Mode Set

	MSB							LSB
Code:	0	0	0	D	D	K	K	K

Where DD is the Display Mode and KKK is the Keyboard Mode.

DD

0	0	8 8-bit character display — Left entry
0	1	16 8-bit character display — Left entry*
1	0	8 8-bit character display — Right entry
1	1	16 8-bit character display — Right entry

For description of right and left entry, see Interface Considerations. Note that when decoded scan is set in keyboard mode, the display is reduced to 4 characters independent of display mode set.

KKK

0	0	0	Encoded Scan Keyboard — 2 Key Lockout
0	0	1	Decoded Scan Keyboard — 2-Key Lockout
0	1	0	Encoded Scan Keyboard — N-Key Rollover
0	1	1	Decoded Scan Keyboard — N-Key Rollover
1	0	0	Encoded Scan Sensor Matrix
1	0	1	Decoded Scan Sensor Matrix
1	1	0	Strobed Input, Encoded Display Scan
1	1	1	Strobed Input, Decoded Display Scan

Program Clock

Code:	0	0	1	P	P	P	P	P

Where PPPPP is the prescaler value 2 to 31. The programmable prescaler divides the external clock by PPPPP to get the basic internal frequency. Choosing a divisor that yields 100 KHz will give the specified scan and debounce times. Default after a reset pulse (but not a program clear) is 31.

Read FIFO/Sensor RAM

Code:	0	1	0	AI	X	A	A	A	X = Don't Care

Where AI is the Auto-Increment flag for the Sensor RAM and AAA is the row that is going to be read by the CPU. AI and AAA are used only if the mode is set to Sensor Matrix. This command is used to specify that the source of data reads ($\overline{CS} \bullet RD \bullet \overline{A_0}$) by the CPU is the FIFO/Sensor RAM. No additional commands are necessary as long as

Default after reset.

data is desired from the FIFO/Sensor RAM. Another command is necessary if reading is desired from a different row than has been selected. If AI is a one, the row select counter will be incremented after each read so the next read will be from the next Sensor RAM row.

In the Auto Increment mode for reading data from the FIFO/Sensor RAM, each read advances the address by one so that the next read is from the next character. This Auto Incrementing has no effect on the display.

Read Display RAM

Code:

0	1	1	AI	A	A	A	A

Where AI is the Auto-Increment flag for the Display RAM and AAAA is the character that the CPU is going to read next. Since the CPU uses the same counter for reading and writing, this command also sets the next write location and Auto-Increment mode. This command is used to specify the display RAM as the data source for CPU data reads. If AI is set, the character address will be incremented after each read (or write) so that the next read (or write) will be from (to) the next character.

Write Display RAM

Code:

1	0	0	AI	A	A	A	A

Where AI is the Auto-Increment flag for the Display RAM and AAAA is the character that the CPU is going to write next. The addressing and Auto-Increment are identical to Read Display RAM. The difference is that Write Display RAM does not affect the source of CPU reads. The CPU will read from whichever RAM (Display or FIFO/Sensor) was last specified. This command will, however, change the location the next Display RAM read will be from if that source was specified.

Display Write Inhibit/Blanking

Code:

1	0	1	X	IW	IW	BL	BL
				A	B	A	B

Where IW is Inhibit Writing (nibble A or B) and BL is Blanking (nibble A or B). If the display is being used as a dual 4-bit display, then it is necessary to mask one of the 4-bit halves so that entries to the Display from the CPU do not affect the other half. The IW flags allow the programmer to do this. It is also useful to be able to blank either half when that half is not to be displayed. The BL flags blank the display. The next command sets the output code to be used as a "blank". Default after reset is all zeros. Note that to blank a display formatted as a single 8-bit output, it is necessary to set both BL flags to entirely blank the display. A "1" sets the flag. Reissuing the command with a "0" resets the flag.

Clear

Code:

1	1	0	C_D	C_D	C_D	C_F	C_A

Where C_D is Clear Display, C_F is Clear FIFO Status (including interrupt), and C_A is Clear All. C_D is used to clear all positions of the Display RAM to a programmable code. All ones, all zeros and hexadecimal 20 are possible. The 2 least significant bits of C_D are also used to specify the blanking code (see below).

C_D C_D	C_D	
0	X	All Zeros (X = Don't Care)
1	0	AB = Hex 20 (0010 0000)
1	1	All Ones

Enable clear display when = 1 (or by C_A = 1)

Clearing the display takes approximately 160 μs. During this time the CPU cannot write to the Display RAM. The MSB of the FIFO status word will be set during this time. C_F set the FIFO status to empty and resets the interrupt output line. After execution of a clear command with C_F set, the Sensor Matrix mode RAM pointer will be set to row 0.

C_A has the combined effect of C_D and C_F. C_A uses the C_D clearing code to determine how to clear the Display RAM. C_A also resets the internal timing chain to resynchronize it.

End Interrupt/Error Mode Set

Code:

1	1	1	E	X	X	X	X

X = Don't care.

For the sensor matrix modes this command lowers the IRQ line and enables further writing into RAM. (The IRQ line would have been raised upon the detection of a change in a sensor value. This would have also inhibited further writing into the RAM until reset).

For the N-key rollover mode — if the E bit is programmed to "1" the chip will operate in the special Error mode. (For further details, see Interface Considerations Section.)

Status Word

The status word contains the FIFO status, error, and display unavailable signals. This word is read by the CPU when A_0 is high and $\overline{CS}$ and $\overline{RD}$ are low. See Interface Considerations for more detail on status word.

Data Read

Data is read when A_0, $\overline{CS}$ and $\overline{RD}$ are all low. The source of the data is specified by the Read FIFO or Read Display commands. The trailing edge of $\overline{RD}$ will cause the address of the RAM being read to be incremented if the Auto-Increment flag is set. FIFO reads always increment (if no error occurs) independent of AI.

Data Write

Data that is written with A_0, $\overline{CS}$ and $\overline{WR}$ low is always written to the Display RAM. The address is specified by the latest Read Display or Write Display command. Auto-Incrementing on the rising edge of $\overline{WR}$ occurs if AI set by the latest display command.

INTERFACE CONSIDERATIONS

Scanned Keyboard Mode, 2-Key Lockout

There are three possible combinations of conditions that can occur during debounce scanning. When a key is depressed, the debounce logic is set. Other depressed keys are looked for during the next two scans. If none are encountered, it is a single key depression and the key position is entered into the FIFO along with the status of CNTL and SHIFT lines. If the FIFO was empty, IRQ will be set to signal the CPU that there is an entry in the FIFO. If the FIFO was full, the key will not be entered and the error flag will be set. If another closed switch is encountered, no entry to the FIFO can occur. If all other keys are released before this one, then it will be entered to the FIFO. If this key is released before any other, it will be entirely ignored. A key is entered to the FIFO only once per depression, no matter how many keys were pressed along with it or in what order they were released. If two keys are depressed within the debounce cycle, it is a simultaneous depression. Neither key will be recognized until one key remains depressed alone. The last key will be treated as a single key depression.

Scanned Keyboard Mode, N-Key Rollover

With N-key Rollover each key depression is treated independently from all others. When a key is depressed, the debounce circuit waits 2 keyboard scans and then checks to see if the key is still down. If it is, the key is entered into the FIFO. Any number of keys can be depressed and another can be recognized and entered into the FIFO. If a simultaneous depression occurs, the keys are recognized and entered according to the order the keyboard scan found them.

Scanned Keyboard — Special Error Modes

For N-key rollover mode the user can program a special error mode. This is done by the "End Interrupt/Error Mode Set" command. The debounce cycle and key-validity check are as in normal N-key mode. If during a single debounce cycle, two keys are found depressed, this is considered a simultaneous multiple depression, and sets an error flag. This flag will prevent any further writing into the FIFO and will set interrupt (if not yet set). The error flag could be read in this mode by reading the FIFO STATUS word. (See "FIFO STATUS" for further details.) The error flag is reset by sending the normal CLEAR command with $C_F = 1$.

Sensor Matrix Mode

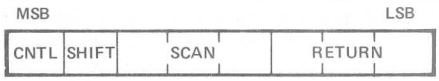
In Sensor Matrix mode, the debounce logic is inhibited. The status of the sensor switch is inputted directly to the Sensor RAM. In this way the Sensor RAM keeps an image of the state of the switches in the sensor matrix. Although debouncing is not provided, this mode has the advantage that the CPU knows how long the sensor was closed and when it was released. A keyboard mode can only indicate a validated closure. To make the software easier, the designer should functionally group the sensors by row since this is the format in which the CPU will read them.

The IRQ line goes high if any sensor value change is detected at the end of a sensor matrix scan. The IRQ line is cleared by the first data read operation if the Auto-Increment flag is set to zero, or by the End Interrupt command if the Auto-Increment flag is set to one.

Note: Multiple changes in the matrix Addressed by ($SL_{0-3} = 0$) may cause multiple interrupts. ($SL_0 = 0$ in the Decoded Mode). Reset may cause the 8279 to see multiple changes.

Data Format

In the Scanned Keyboard mode, the character entered into the FIFO corresponds to the position of the switch in the keyboard plus the status of the CNTL and SHIFT lines (non-inverted). CNTL is the MSB of the character and SHIFT is the next most significant bit. The next three bits are from the scan counter and indicate the row the key was found in. The last three bits are from the column counter and indicate to which return line the key was connected.



SCANNED KEYBOARD DATA FORMAT

In Sensor Matrix mode, the data on the return lines is entered directly in the row of the Sensor RAM that corresponds to the row in the matrix being scanned. Therefore, each switch postion maps directly to a Sensor RAM position. The SHIFT and CNTL inputs are ignored in this mode. Note that switches are not necessarily the only thing that can be connected to the return lines in this mode. Any logic that can be triggered by the scan lines can enter data to the return line inputs. Eight multiplexed input ports could be tied to the return lines and scanned by the 8279.



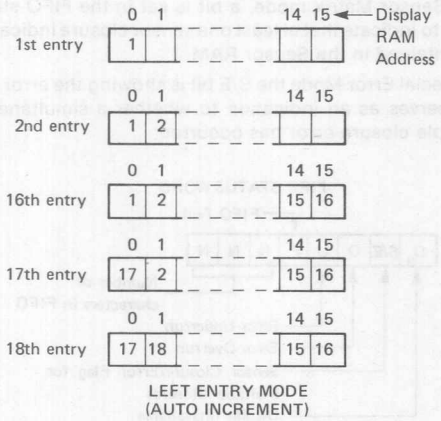
In Strobed Input mode, the data is also entered to the FIFO from the return lines. The data is entered by the rising edge of a CNTL/STB line pulse. Data can come from another encoded keyboard or simple switch matrix. The return lines can also be used as a general purpose strobed input.



Display

Left Entry

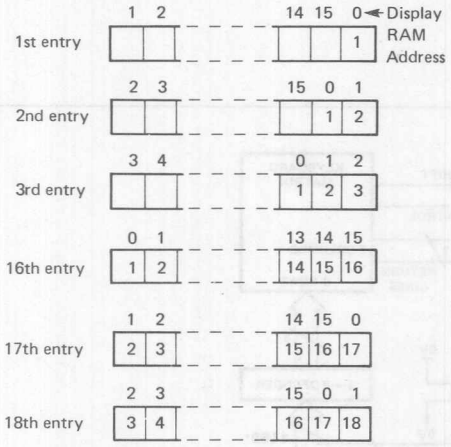
Left Entry mode is the simplest display format in that each display position directly corresponds to a byte (or nibble) in the Display RAM. Address 0 in the RAM is the left-most display character and address 15 (or address 7 in 8 character display) is the right most display character. Entering characters from position zero causes the display to fill from the left. The 17th (9th) character is entered back in the left most position and filling again proceeds from there.



LEFT ENTRY MODE
(AUTO INCREMENT)

Right Entry

Right entry is the method used by most electronic calculators. The first entry is placed in the right most display character. The next entry is also placed in the right most character after the display is shifted left one character. The left most character is shifted off the end and is lost.

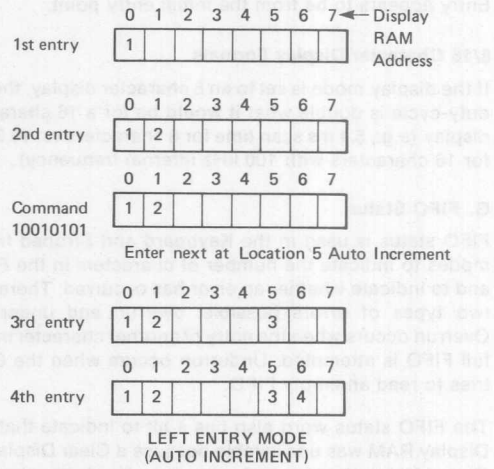


RIGHT ENTRY MODE
(AUTO INCREMENT)

Note that now the display position and register address do not correspond. Consequently, entering a character to an arbitrary position in the Auto Increment mode may have unexpected results. Entry starting at Display RAM address 0 with sequential entry is recommended.

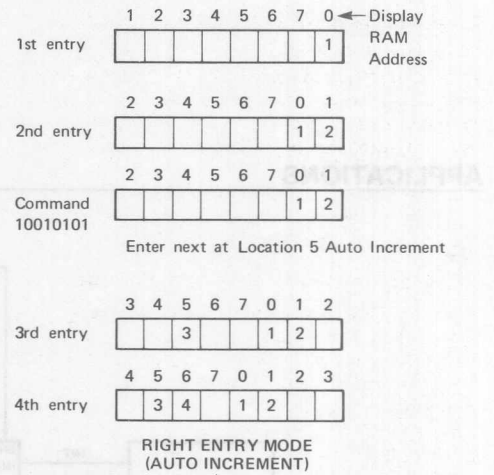
Auto Increment

In the Left Entry mode, Auto Incrementing causes the address where the CPU will next write to be incremented by one and the character appears in the next location. With non-Auto Incrementing the entry is both to the same RAM address and display position. Entry to an arbitrary address in the Auto Increment mode has no undesirable side effects and the result is predictable:



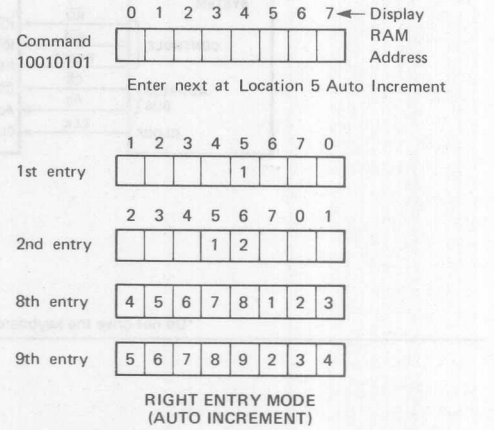
LEFT ENTRY MODE
(AUTO INCREMENT)

In the Right Entry mode, Auto Incrementing and non Incrementing have the same effect as in the Left Entry except if the address sequence is interrupted:



RIGHT ENTRY MODE
(AUTO INCREMENT)

Starting at an arbitrary location operates as shown below:



RIGHT ENTRY MODE
(AUTO INCREMENT)

Entry appears to be from the initial entry point.

8/16 Character Display Formats

If the display mode is set to an 8 character display, the on duty-cycle is double what it would be for a 16 character display (e.g., 5.1 ms scan time for 8 characters vs. 10.3 ms for 16 characters with 100 kHz internal frequency).

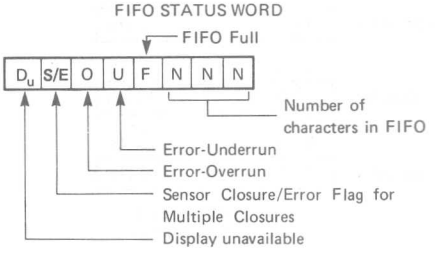
G. FIFO Status

FIFO status is used in the Keyboard and Strobed Input modes to indicate the number of characters in the FIFO and to indicate whether an error has occurred. There are two types of errors possible: overrun and underrun. Overrun occurs when the entry of another character into a full FIFO is attempted. Underrun occurs when the CPU tries to read an empty FIFO.

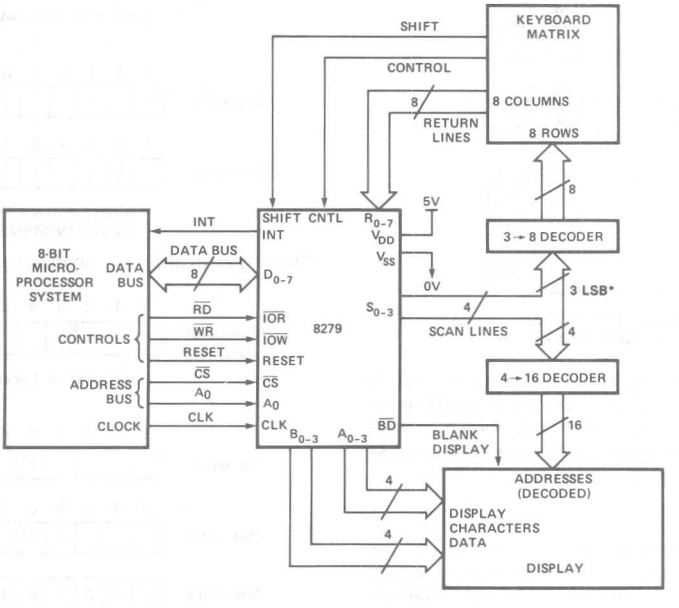
The FIFO status word also has a bit to indicate that the Display RAM was unavailable because a Clear Display or Clear All command had not completed its clearing operation.

In a Sensor Matrix mode, a bit is set in the FIFO status word to indicate that at least one sensor closure indication is contained in the Sensor RAM.

In Special Error Mode the S/E bit is showing the error flag and serves as an indication to whether a simultaneous multiple closure error has occurred.



APPLICATIONS



*Do not drive the keyboard decoder with the MSB of the scan lines.

PRELIMINARY
Notice: This is a product in development. Some parameters may be subject to change.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature 0°C to 70°C
Storage Temperature -65°C to 125°C
Voltage on any Pin with
Respect to Ground -0.5V to +7V
Power Dissipation 1 Watt

**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{SS} = 0V, Note 1

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{IL1}	Input Low Voltage for Return Lines	-0.5	1.4	V	
V_{IL2}	Input Low Voltage for All Others	-0.5	0.8	V	
V_{IH1}	Input High Voltage for Return Lines	2.2		V	
V_{IH2}	Input High Voltage for All Others	2.0		V	
V_{OL}	Output Low Voltage		0.45	V	Note 2
V_{OH}	Output High Voltage on Interrupt Line	3.5		V	Note 3
I_{IL1}	Input Current on Shift, Control and Return Lines		+10 -100	μA μA	$V_{IN} = V_{CC}$ V_{IN} = 0V
I_{IL2}	Input Leakage Current on All Others		±10	μA	$V_{IN} = V_{CC}$ to 0V
I_{OFL}	Output Float Leakage		±10	μA	$V_{OUT} = V_{CC}$ to 0V
I_{CC}	Power Supply Current		120	mA	

Notes:
1. 8279, V_{CC} = +5V ±5%; 8279-5, V_{CC} = +5V ±10%.
2. 8279, I_{OL} = 1.6mA; 8279-5, I_{OL} = 2.2mA.
3. 8279, I_{OH} = -100μA; 8279-5, I_{OH} = -400μA.

CAPACITANCE

SYMBOL	TEST	TYP.	MAX.	UNIT	TEST CONDITIONS
C_{in}	Input Capacitance	5	10	pF	$V_{in}=V_{CC}$
C_{out}	Output Capacitance	10	20	pF	$V_{out}=V_{CC}$

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{SS} = 0V, (Note 1)

Bus Parameters

Read Cycle:

Symbol	Parameter	8279 Min.	8279 Max.	8279-5 Min.	8279-5 Max.	Unit
t_{AR}	Address Stable Before $\overline{READ}$	50		0		ns
t_{RA}	Address Hold Time for $\overline{READ}$	5		0		ns
t_{RR}	$\overline{READ}$ Pulse Width	420		250		ns
t_{RD}[2]	Data Delay from $\overline{READ}$		300		150	ns
t_{AD}[2]	Address to Data Valid		450		250	ns
t_{DF}	$\overline{READ}$ to Data Floating	10	100	10	100	ns
t_{RCY}	Read Cycle Time	1		1		μs

Write Cycle:

Symbol	Parameter	8279 Min.	8279 Max.	8279-5 Min.	8279-5 Max.	Unit
t_{AW}	Address Stable Before $\overline{WRITE}$	50		0		ns
t_{WA}	Address Hold Time for $\overline{WRITE}$	20		0		ns
t_{WW}	$\overline{WRITE}$ Pulse Width	400		250		ns
t_{DW}	Data Set Up Time for $\overline{WRITE}$	300		150		ns
t_{WD}	Data Hold Time for $\overline{WRITE}$	40		0		ns

Notes:
1. 8279, V_{CC} = +5V ±5%; 8279-5, V_{CC} = +5V ±10%.
2. 8279, C_L = 100pF; 8279-5, C_L = 150pF.

Other Timings:

Symbol	Parameter	8279 Min.	8279 Max.	8279-5 Min.	8279-5 Max.	Unit
$t_{\phi W}$	Clock Pulse Width	230		120		nsec
t_{CY}	Clock Period	500		320		nsec

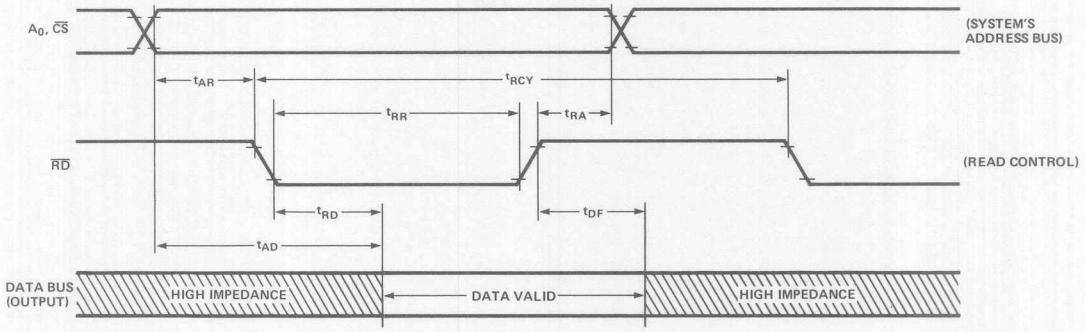
Keyboard Scan Time:	5.1 msec	Digit-on Time:	480 μsec
Keyboard Debounce Time:	10.3 msec	Blanking Time:	160 μsec
Key Scan Time:	80 μsec	Internal Clock Cycle:	10 μsec
Display Scan Time:	10.3 msec		

Input Waveforms For A.C. Tests

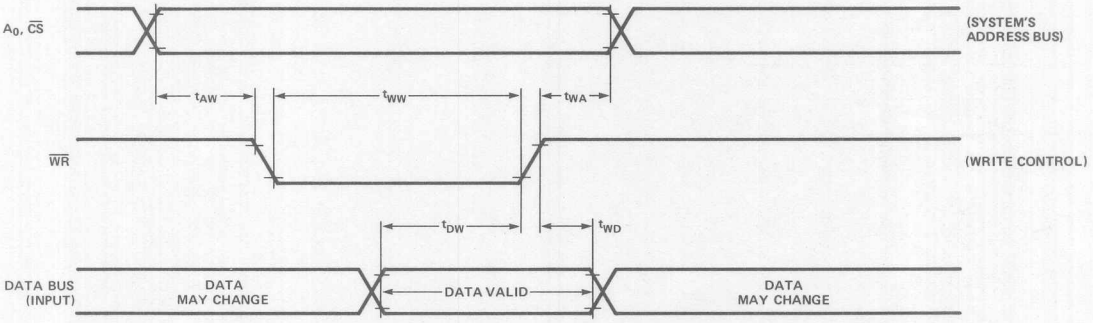


WAVEFORMS

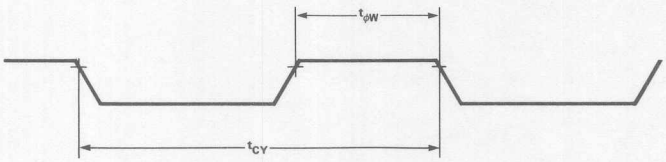
Read Operation

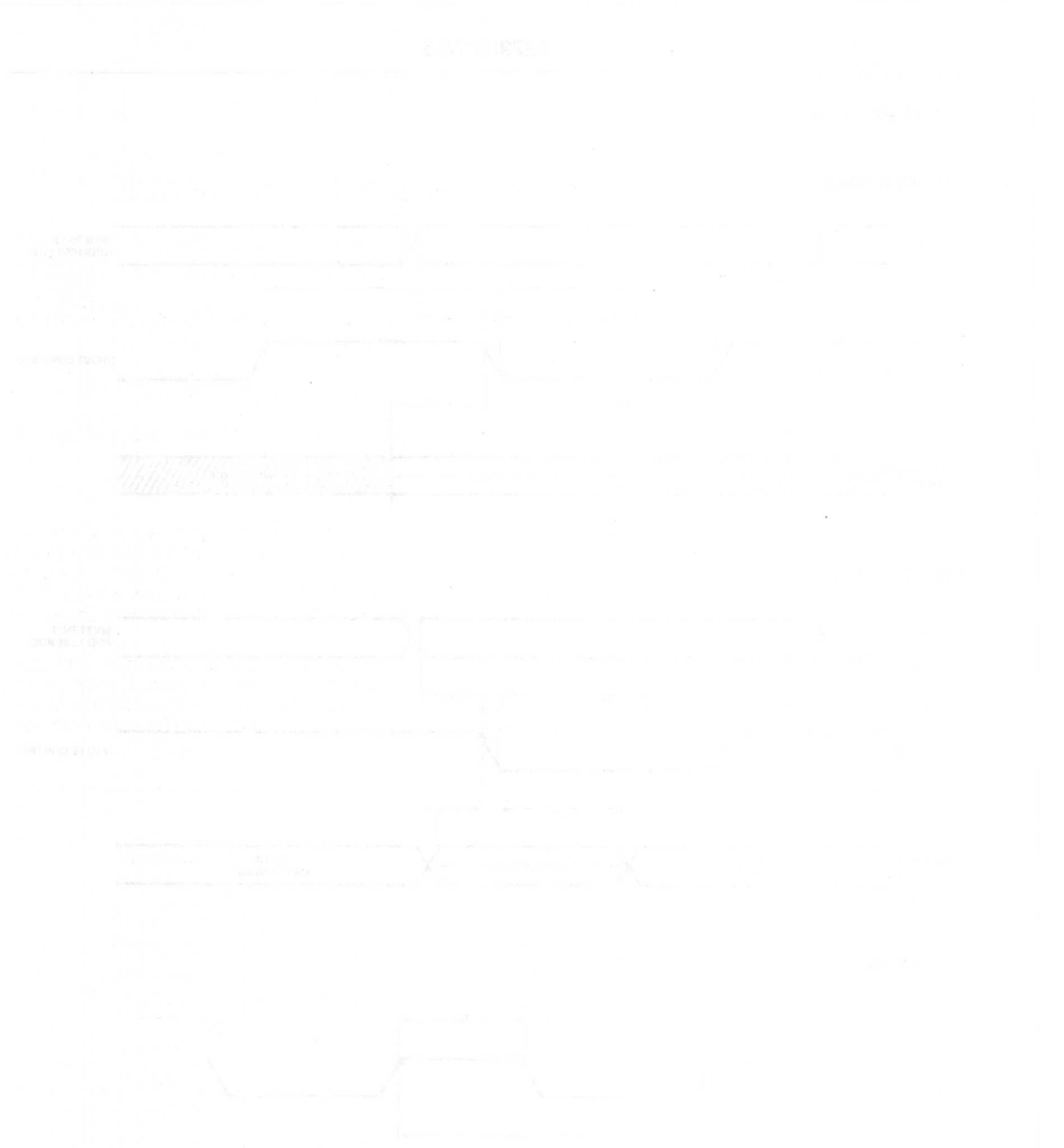


Write Operation



Clock Input





PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

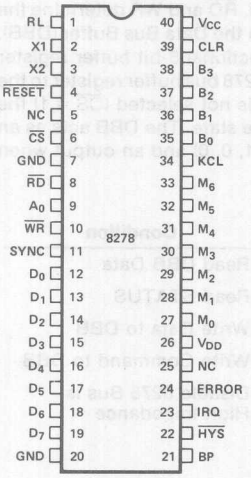
8278
PROGRAMMABLE KEYBOARD INTERFACE

- **Simultaneous Keyboard and Display Operations**
- **Interface Signals for Contract and Capacitive Coupled Keyboards**
- **128-Key Scanning Logic**
- **10.7 msec Matrix Scan Time for 128 Keys and 6 MHz Clock**
- **8-Character Keyboard FIFO**
- **N-Key Rollover with Programmable Error Mode on Multiple New Closures**
- **16- or 18-Character 7-Segment Display Interface**
- **Right or Left Entry Display RAM**
- **Depress/Release Mode Programmable**
- **Interrupt Output on Key Entry**

The Intel® 8278 is a general purpose programmable keyboard and display interface device designed for use with 8-bit microprocessors such the MDS-80™ and MCS-85™. The keyboard portion can provide a scanned interface to 128-key contact or capacitive-coupled keyboards. The keys are fully debounced with N-key rollover and programmable error generation on multiple new key closures. Keyboard entries are stored in an 8-character FIFO with overrun status indication when more than 8 characters are entered. Key entries set an interrupt request output to the master CPU.

The display portion of the 8278 provides a scanned display interface for LED, incandescent, and other popular display technologies. Both numeric displays and simple indicators may be used. The 8278 has a 16X4 display RAM which can be loaded or interrogated by the CPU. Both right entry calculator and left entry typewriter display formats are possible. Both read and write of the display RAM can be done with auto-increment of the display RAM address.

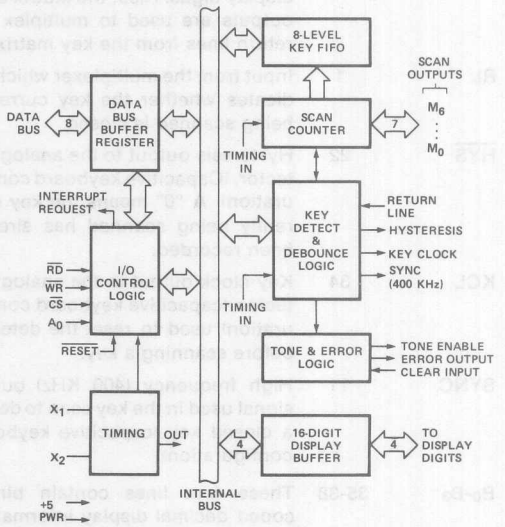
PIN CONFIGURATION



PIN NAMES

D_7-D_0	DATA BUS
$\overline{RD}$, $\overline{WR}$	READ, WRITE STROBES
$\overline{CS}$	CHIP SELECT
A_0	CONTROL/DATA SELECT
$\overline{RESET}$	RESET INPUT
X_1, X_2	FREQ. REFERENCE INPUT
SYNC	HIGH FREQUENCY OUTPUT CLOCK
RL	KEYBOARD RETURN LINE
CLR	CLEAR ERROR
KCL	KEY CLOCK
M_6–M_0	MATRIX SCAN LINES
B_3-B_0	DISPLAY OUTPUTS
ERROR	ERROR SIGNAL
$\overline{IRQ}$	INTERRUPT REQUEST
$\overline{HYS}$	HYSTERESIS
BP	TONE ENABLE

BLOCK DIAGRAM



© Intel Corporation, 1978

PIN DESCRIPTION

The 8278 is packaged in a 40-pin DIP. The following is a brief functional description of each pin.

Signal	Pin No.	Description
D_0-D_7	12-19	Three-state, bi-directional data bus lines used to transfer data and commands between the CPU and the 8278.
$\overline{WR}$	10	Write strobe which enables the master CPU to write data and commands between the CPU and the 8278.
$\overline{RD}$	8	Read strobe which enables the master CPU to read data and status from the 8278 internal registers.
$\overline{CS}$	6	Chip select input used to enable reading and writing to the 8278.
A_0	9	Address input used by the CPU to indicate control or data.
$\overline{RESET}$	4	A low signal on this pin resets the 8278.
X_1, X_2	2,3	Inputs for crystal, L-C or external timing signal to determine internal oscillator frequency.
IRQ	23	Interrupt Request Output to the master CPU. In the keyboard mode the IRQ line goes low with each FIFO read and returns high if there is still information in the FIFO or an ERROR has occurred.
M_0-M_6	27-33	Matrix scan outputs. These outputs control a decoder which scans the key matrix columns and the 16 display digits. Also, the Matrix scan outputs are used to multiplex the return lines from the key matrix.
RL	1	Input from the multiplexer which indicates whether the key currently being scanned is closed.
$\overline{HYS}$	22	Hysteresis output to the analog detector. (Capacitive keyboard configuration). A "0" means the key currently being scanned has already been recorded.
KCL	34	Key clock output to the analog detector (capacitive keyboard configuration) used to reset the detector before scanning a key.
SYNC	11	High frequency (400 KHz) output signal used in the key scan to detect a closed key (capacitive keyboard configuration).
B_0-B_3	35-38	These four lines contain binary coded decimal display information synchronized to the keyboard column scan. The outputs are for multiplexed digital displays.
ERROR	24	Error signal. This line is high whenever two new key closures are detected during a single scan or when too many characters are entered into the keyboard FIFO. It is reset by a system RESET pulse or by a "1" input on the CLR pin or by the CLEAR ERROR command.
CLR	39	Input used to clear an ERROR condition in the 8278.
BP	21	Tone enable output. This line is high for 10ms following a valid key closure; it is set high and remains high during an ERROR condition.
V_{CC}, V_{DD}	40,26	+5 volt power input: +5V ± 10%.
GND	20,7	Signal ground.

PRINCIPLES OF OPERATION

The following is a description of the major elements of the Programmable Keyboard/Display interface device. Refer to the block diagram in Figure 1.

I/O Control and Data Buffers

The I/O control section uses the $\overline{CS}$, A_0, $\overline{RD}$, and $\overline{WR}$ lines to control data flow to and from the various internal registers and buffers (see Table 1). All data flow to and from the 8278 is enabled by $\overline{CS}$. The 8-bits of information being transferred by the CPU is identified by A_0. A logic one means information is command or status. A logic zero means the information is data. $\overline{RD}$ and $\overline{WR}$ determine the direction of data flow through the Data Bus Buffer (DBB). The DBB register is a bi-directional 8-bit buffer register which connects the internal 8278 bus buffer register to the external bus. When the chip is not selected ($\overline{CS}$ = 1) the DBB is in the high impedance state. The DBB acts as an input when ($\overline{RD}$, $\overline{WR}$, $\overline{CS}$) = (1, 0, 0) and an output when ($\overline{RD}$, $\overline{WR}$, $\overline{CS}$) = (0, 1, 0).

$\overline{CS}$	A_0	$\overline{WR}$	$\overline{RD}$	Condition
0	0	1	0	Read DBB Data
0	1	1	0	Read STATUS
0	0	0	1	Write Data to DBB
0	1	0	1	Write Command to DBB
1	X	X	X	Disable 8278 Bus is High Impedance

Scan Counter

The scan counter provides the timing to scan the keyboard and display. The four MSB's (M_3-M_6) scan the display digits and provide column scan to the keyboard via a 4 to 16 decoder. The three LSB's (M_0-M_2) are used to multiplex the row return lines into the 8278.

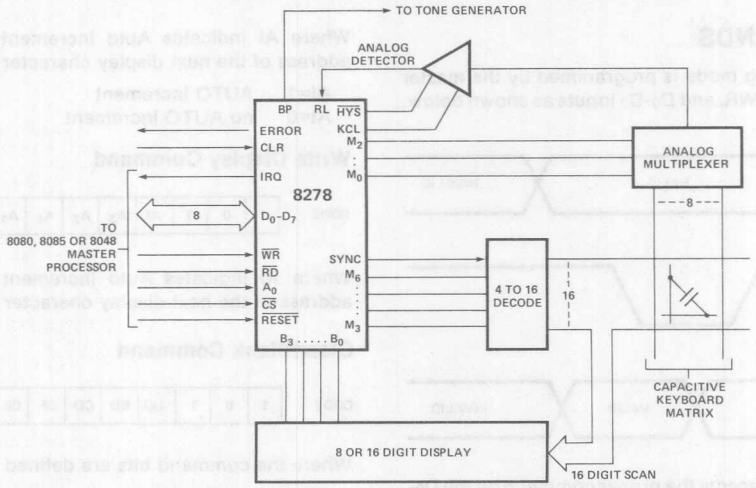


Figure 1. System Configuration for Capacitive-Coupled Keyboard

Keyboard Debounce and Control

The 8278 system configuration is shown in Figure 2. The rows of the matrix are scanned and the outputs are multiplexed by the 8278. When a key closure is detected, the debounce logic waits about 12 msec to check if the key remains closed. If it does, the address of the key in the matrix is transferred into a FIFO buffer.

FIFO and FIFO Status

The 8278 contains an 8X8 FIFO character buffer. Each new entry is written into a successive FIFO location and each is then read out in the order of entry. A FIFO status register keeps track of the number of characters in the FIFO and whether it is full or empty. Too many reads or key entries will be recognized as an error. The status can be read by a $\overline{RD}$ with $\overline{CS}$ low and A_0 high. The status logic also provides a IRQ signal to the master processor whenever the FIFO is not empty.

Display Address Registers and Display RAM

The Display Address registers hold the address of the word currently being written or read by the CPU and the two 4-bit nibbles being displayed. The read/write addresses are programmed by CPU command. They also can be set to auto increment after each read or write. The display RAM can be directly read by the CPU after the correct mode and address is set. Data entry to the display can be set to either left or right entry.

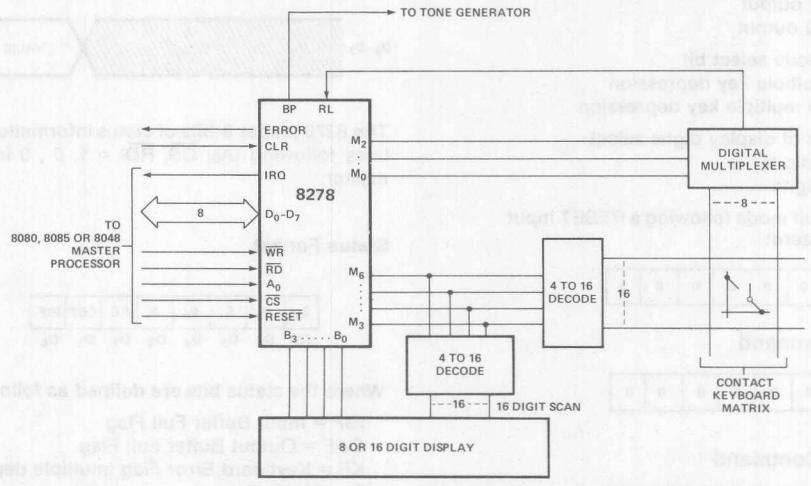
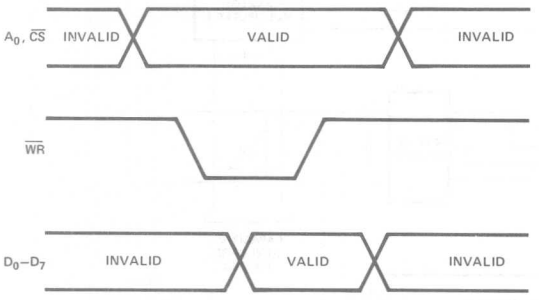


Figure 2. System Configuration for Contact Keyboard

8278 COMMANDS

The 8278 operating mode is programmed by the master CPU using the A_0, WR, and D_0-D_7 inputs as shown below:



The master CPU presents the proper command on the D_0-D_7 data lines with A_0=1 and then sends a $\overline{WR}$ pulse. The command is latched by the 8278 on the rising edge of the $\overline{WR}$ and is decoded internally to set the proper operating mode.

COMMAND SUMMARY

Keyboard/Display Mode Set

CODE	0	0	0	N	E	I	D	K

where the mode set bits are defined as follows:

K — the keyboard mode select bit
0 — normal key entry mode
1 — special function mode: Entry on key closure and on key release

D — the display entry mode select bit
0 — left display entry
1 — right display entry

I — the interrupt request (IRQ) output enable bit.
0 — enable IRQ output
1 — disable IRQ output

E — the error mode select bit
0 — error on multiple key depression
1 — no error on multiple key depression

N — the number of display digits select
0 — 16 display digits
1 — 8 display digits

NOTE: The default mode following a RESET input is all bits zero:

0	0	0	0	0	0	0	0

Read FIFO Command

CODE	0	1	0	0	0	0	0	0

Read Display Command

CODE	0	1	1	AI	A_3	A_2	A_1	A_0

Where AI indicates Auto Increment and A_3-A_0 is the address of the next display character to be read out.

AI=1 AUTO increment
AI=0 no AUTO increment

Write Display Command

CODE	1	0	0	AI	A_3	A_2	A_1	A_0

Where AI indicates Auto Increment and A_3-A_0 is the address of the next display character to be written.

Clear/Blank Command

CODE	1	0	1	UD	BD	CD	CF	CE

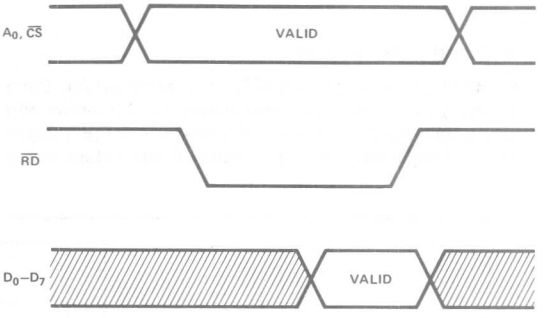
Where the command bits are defined as follows:

CE = Clear ERROR
CF = Clear FIFO
CD = Clear Display to all High
BD = Blank Display to all High
UD = Unblank Display

The display is cleared and blanked following a Reset.

8278 Status Read

The status register in the 8278 can be read by the master CPU using the A_0, $\overline{RD}$, and D_0-D_7 inputs as shown below:



The 8278 places 8-bits of status information on the D_0-D_7 lines following (A_0, $\overline{CS}$, $\overline{RD}$) = 1, 0 , 0 inputs from the master.

Status Format

S_3	S_2	S_1	S_0	B	KE	OBF	IBF
D_7	D_6	D_5	D_4	D_3	D_2	D_1	D_0

Where the status bits are defined as follows:

IBF = Input Buffer Full Flag
OBF = Output Buffer Full Flag
KE = Keyboard Error Flag (multiple depression)
B = BUSY Flag
S_3-S_0 = FIFO Status

Status Description

The S_3-S_0 status bits indicate the number of entries (0 to 8) in the 8-level FIFO. A FIFO overrun will lock status at 1111. The overrun condition will prevent further key entries until cleared.

A multiple key closure error will set the KE flag and prevent further key entries until cleared.

The IBF and OBF flags signify the status of the 8278 data buffer registers used to transfer information (data, status or commands) to and from the master CPU.

The IBF flag is set when the master CPU writes Data or Commands to the 8278. The IBF flag is cleared by the 8278 during its response to the Data or Command.

The OBF flag is set when the 8278 has output data ready for the master CPU. This flag is cleared by a master CPU Data READ.

The Busy flag in the status register is used as a LOCK-OUT signal to the master processor during response to any command or data write from the master.

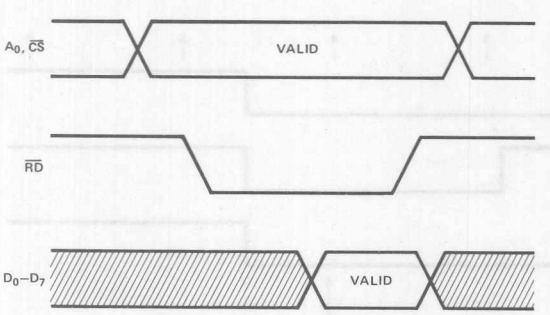
The master must test the Busy flag before each read (during a sequence) to be sure that the 8278 is ready with valid DATA.

The ERROR and TONE outputs from the 8278 are set high for either type of error. Both types of error are cleared by the CLR input, by the CLEAR ERROR command, or by a reset. The FIFO and Display buffers are cleared independently of the Errors.

FIFO status is used to indicate the number of characters in the FIFO and to indicate whether an error has occurred. Overrun occurs when the entry of another character into a full FIFO is attempted. Underrun occurs when the CPU tries to read an empty FIFO. The character read will be the last one entered. FIFO status will remain at 0000 and the error condition will not be set.

8278 Data Read

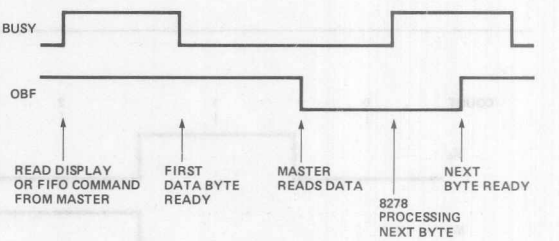
The master CPU can read DATA from the 8278 FIFO or Display buffers by using the A_0, $\overline{RD}$, and D_0-D_7 inputs as follows:



The master sends a $\overline{RD}$ pulse with A_0=0 and CS=0 and the 8278 responds by outputing data on lines D_0-D_7. The data is strobed by the trailing edge of $\overline{RD}$.

Data Read Sequence

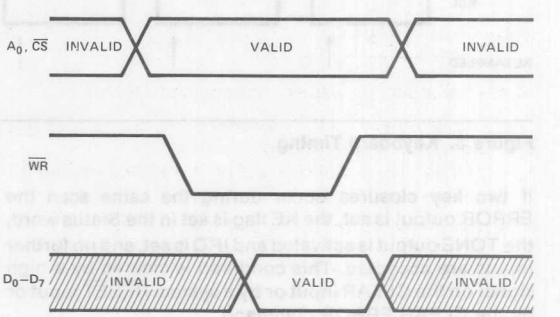
Before reading data, the master CPU must send a command to select FIFO or Display data. Following the command, the master must read STATUS and test the BUSY flag and the OBF flag to verify that the 8278 has responded to the previous command. A typical DATA READ sequence is as follows:



After the first read following a Read Display or Read FIFO command, successive reads may occur as soon as OBF rises.

8278 Data Write

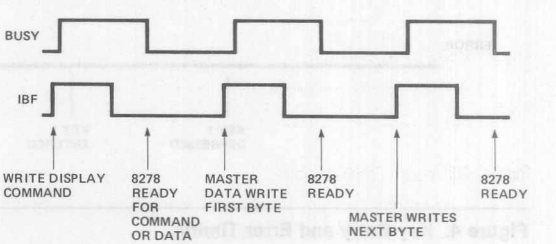
The master CPU can write DATA to the 8278 Display buffers by using the A_0, $\overline{WR}$ and D_0-D_7 inputs as follows:



The master CPU presents the Data on the D_0-D_7 lines with A_0=0 and then sends a $\overline{WR}$ pulse. The data is latched by the 8278 on the rising edge of $\overline{WR}$.

Data Write Sequence

Before writing data to the 8278, the master CPU must first send a command to select the desired display entry mode and to specify the address of the next data byte. Following the commands, the master must read STATUS and test the BUSY flag (B) and IBF flag to verify that the 8278 has responded. A typical sequence is shown below:



INTERFACE CONSIDERATIONS

Scanned Keyboard Mode

With N-key rollover each key depression is treated independently from all others. When a key is depressed the debounce logic waits for a full scan of 128 keys and then checks to see if the key is still down. If it is, the key is entered into the FIFO.

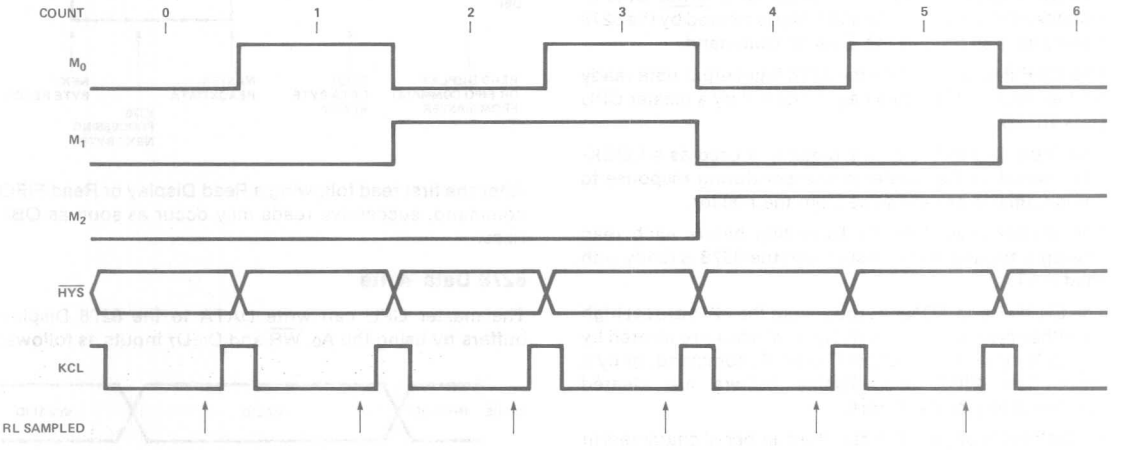


Figure 3. Keyboard Timing

If two key closures occur during the same scan the ERROR output is set, the KE flag is set in the Status word, the TONE output is activated and IRQ is set, and no further inputs are accepted. This condition is cleared by a high signal on the CLEAR input or by a system $\overline{RESET}$ input or by the CLEAR ERROR command.

In the special function mode both the key closure and the key release cause an entry to the FIFO. The release is entered with the MSB=1.

Any key entry triggers the TONE output for 10ms.

The $\overline{HYS}$ and KCL outputs enable the analog multiplexer and detector to be synchronized for interface to capacitive coupled keyboards.

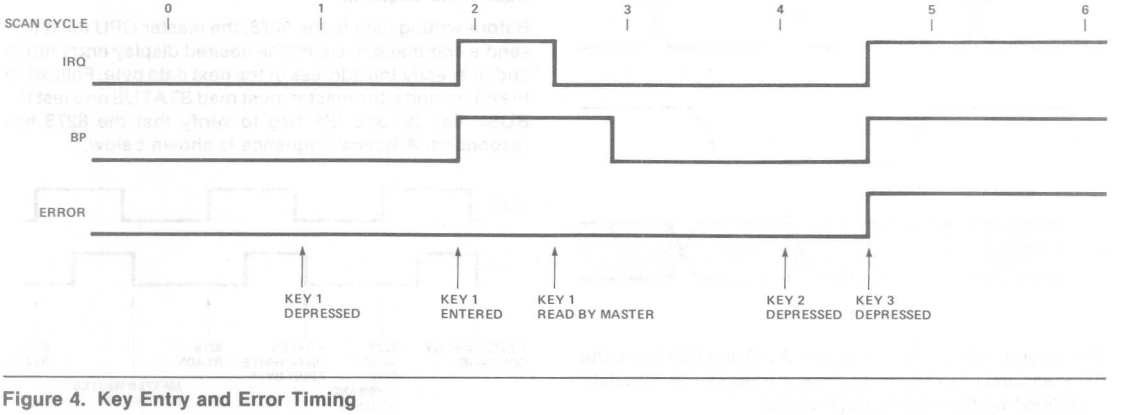


Figure 4. Key Entry and Error Timing

Data Format

In the scanned keyboard mode, the code entered into the FIFO corresponds to the position or address of the switch in the keyboard. The MSB is relevant only for special function keys in which code "0" signifies closure and "1" signifies release. The next four bits are the column count which indicates which column the key was found in. The last three bits are from the row counter.

KEY CODING

BIT	7	6	5	4	3	2	1	0
	SF	M_6	M_5	M_4	M_3	M_2	M_1	M_0

1 FOR SPECIAL FUNCTION MODE AND KEY RELEASED
0 FOR KEY DEPRESSED

Display

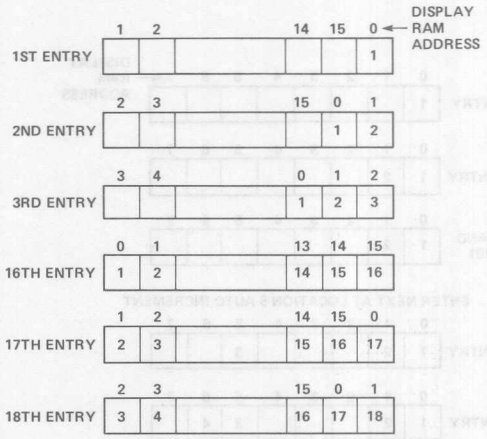
Display data is entered into a 16x4 display register and may be entered from the left, from the right or into specific locations in the display register. A new data character is put out on B_0-B_3 each time the M_6-M_3 lines change (i.e., once every 0.75ms with a 6 MHz crystal). Data is blanked during the time the column select lines change by raising the display outputs. Output data is positive true.

Left Entry

The left entry mode is the simplest display format in that each display position in the display corresponds to a byte (or nibble) in the Display RAM. Address 0 in the RAM is the left-most display character and address 15 is the right-most display character. Entering characters from position zero causes the display to fill from the left. The 17th character is entered back in the left-most position and filling again proceeds from there.

Right Entry

Right entry is the method used by most electronic calculators. The first entry is placed in the right-most display character. The next entry is also placed in the right-most character after the display is shifted left one character. The left-most character is shifted off the end and is lost.



Note that now the display position and register address do not correspond. Consequently, entering a character to an arbitrary position in the Auto Increment mode may have unexpected results. Entry starting at Display RAM address 0 with sequential entry is recommended. A Clear Display command should be given before display data is entered if the number of data characters is not equal to 16 (or 8) in this mode.

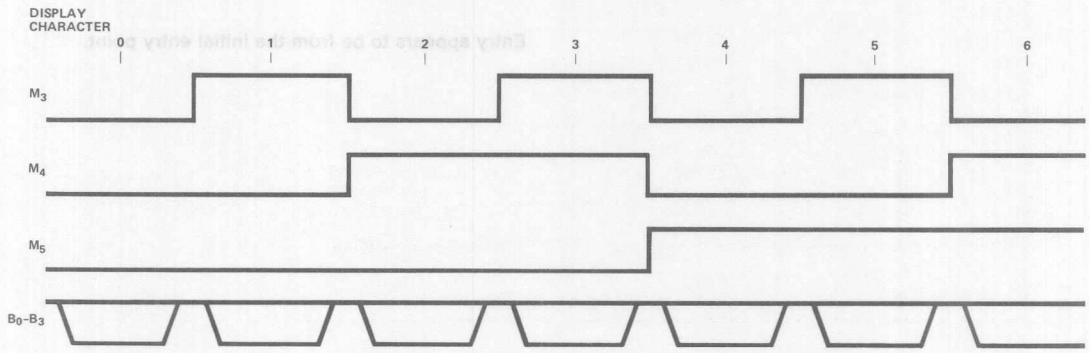
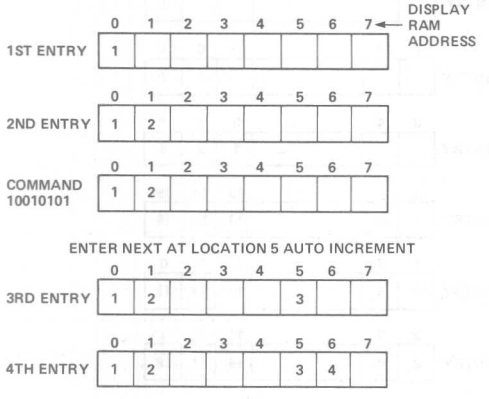


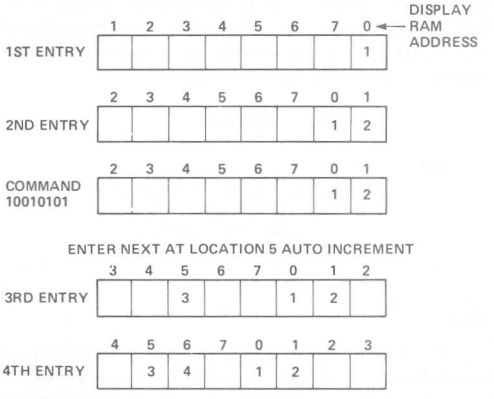
Figure 5. Display Timing

Auto Increment

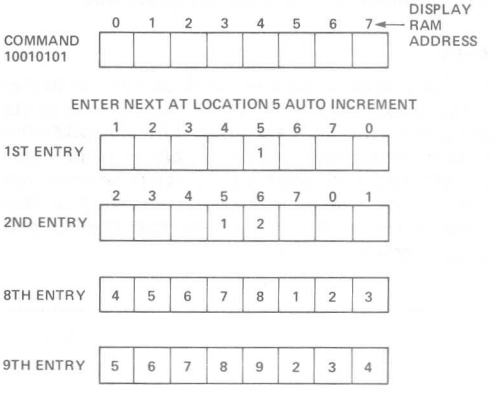
In the Left Entry mode, Auto Incrementing causes the address where the CPU will next write to be incremented by one and the character appears in the next location. With non-Auto Incrementing the entry is both to the same RAM address and display position. Entry to an arbitrary address in the Left Entry — Auto Increment mode has no undesirable side effects and the result is predictable:



In the Right Entry mode, Auto Incrementing and non Incrementing have the same effect as in the Left Entry except that the address sequence is interrupted:



Starting at an arbitrary location operates as shown below:



Entry appears to be from the initial entry point.

PRELIMINARY
Notice: This is a preliminary data sheet. Some parameters may be subject to change.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias 0°C to 70°C
Storage Temperature -65°C to +150°C
Voltage on Any Pin With Respect to Ground 0.5V to +7V
Power Dissipation 1.5 Watt

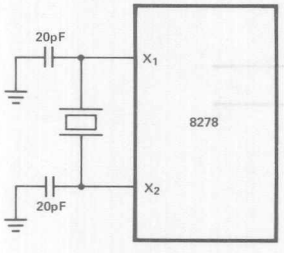
**COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.*

D.C. CHARACTERISTICS

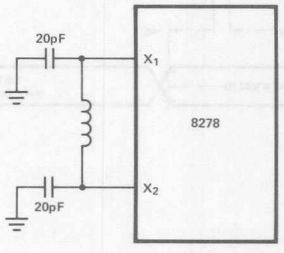
Commercial: T_A = 0°C to 70°C; V_{CC} = +5V ±5%; V_{SS} = 0V

Symbol	Parameter	Min.	Max.	Units	Condition
V_{IL}	Input Low Voltage (All Inputs Except X_1, X_2	-0.5	0.8	V	
V_{IH1}	Input High Voltage (All Inputs Except X_1, X_2, $\overline{RESET}$	2.0	V_{CC}	V	
V_{IH2}	$\overline{RESET}$ High Voltage	3.0	V_{CC}	V	
V_{OL1}	Output Low Voltage (D_0-D_7)		0.45	V	I_{OL} = 2.0mA
V_{OL2}	Output Low Voltage (All Other Outputs)		0.45	V	I_{OL} = 1.6mA
V_{OH1}	Output High Voltage (D_0-D_7)	2.4		V	I_{OH} = -400μA
V_{OH2}	Output High Voltage (All Other Outputs)	2.4		V	I_{OH} = -50μA
I_{IL}	Input Leakage Current (All Inputs Except $\overline{RESET}$)		±10	μA	V_{IN} = V_{CC}
I_{OL}	Output Leakage Current (D_0-D_7)		±10	μA	V_{IN} = V_{SS} + 0.45V or V_{IN} = V_{CC}
I_{DD} + I_{CC}	Total Supply Current		135	mA	V_{CC} = 5.5V
I_{DD}	V_{DD} Supply Current		25	mA	V_{CC} = 5.5V
I_{LI}	Low Input Source Current ($\overline{RESET}$)		0.2	mA	V_{IL} = 0.8V

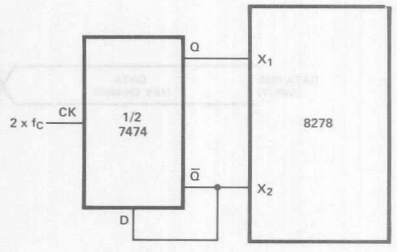
8278 CLOCK OPTIONS



1-6 MHz CRYSTAL



40 μh-130 μh INDUCTOR



EXTERNAL CLOCK

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

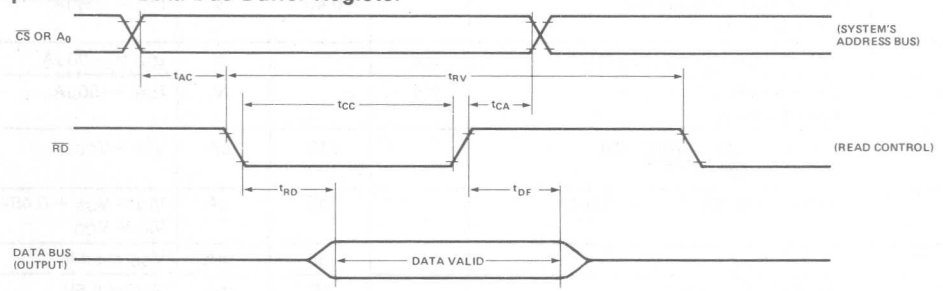
A.C. CHARACTERISTICS

T_A = 0°C to 70°C; V_{CC} = +5V ±10%; V_{SS} = 0V

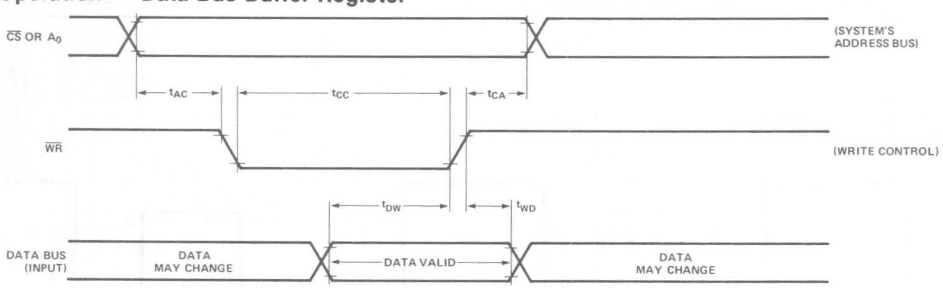
Symbol	Parameter	Min.	Max.	Units	Condition
t_{AC}	Address ($\overline{CS}$, A_0) Setup to Control ($\overline{RD}$, $\overline{WR}$)	0		ns	D_0-D_7, C_L = 150pF
t_{CA}	Address Hold from Control	0		ns	
t_{CC}	Control Pulse Width	250		ns	
t_{DW}	Data in Setup to $\overline{WR}$ T.E.	150		ns	
t_{WD}	Data in Hold After $\overline{WR}$ T.E.	0		ns	
t_{RD}	$\overline{RD}$ L.E. to Data Out Valid		150	ns	
t_{DF}	$\overline{RD}$ T.E. to Data Out Float	10	100	ns	
t_{MCY}	Matrix Cycle Time		10.7	ms	With 6MHz Crystal
t_{RV}	Recovery Time Between Reads and/or Writes	1		μs	

WAVEFORMS

Read Operation — Data Bus Buffer Register



Write Operation — Data Bus Buffer Register



PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

8041/8741
UNIVERSAL PERIPHERAL INTERFACE
8-BIT MICROCOMPUTER

- **Fully Compatibnle with MCS-80™, MCS-85™ and MCS-48™ Microprocessor Families**
- **Single Level Interrupt**
- **8-Bit CPU plus ROM, RAM, I/O, Timer and Clock in a Single Package**
- **Single 5V Supply**
- **Alternative to Custom LSI**
- **Pin Compatible ROM and EPROM Versions**
- **1K x 8 ROM/EPROM, 64 x 8 RAM, 18 Programmable I/O Pins**
- **Asynchronous Data Register for Interface to Master Processor**
- **Expandable I/O**

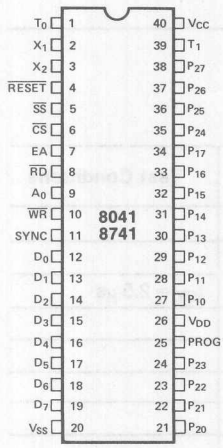
The Intel® 8041/8741 is a general purpose, programmable interface device designed for use with a variety of 8-bit microprocessor systems. It contains a low cost microcomputer with program memory, data memory, 8-bit CPU, I/O ports, timer/counter, and clock in a single 40-pin package. Interface registers are included to enable the UPI device to function as a peripheral controller in MCS-80™, MCS-85™, MCS-48™, and other 8-bit systems.

The UPI-41™ has 1K words of program memory and 64 words of data memory on-chip. To allow full user flexibility the program memory is available as ROM in the 8041 version or as UV-erasable EPROM in the 8741 version. The 8741 and the 8041 are fully pin compatible for easy transition from prototype to production level designs.

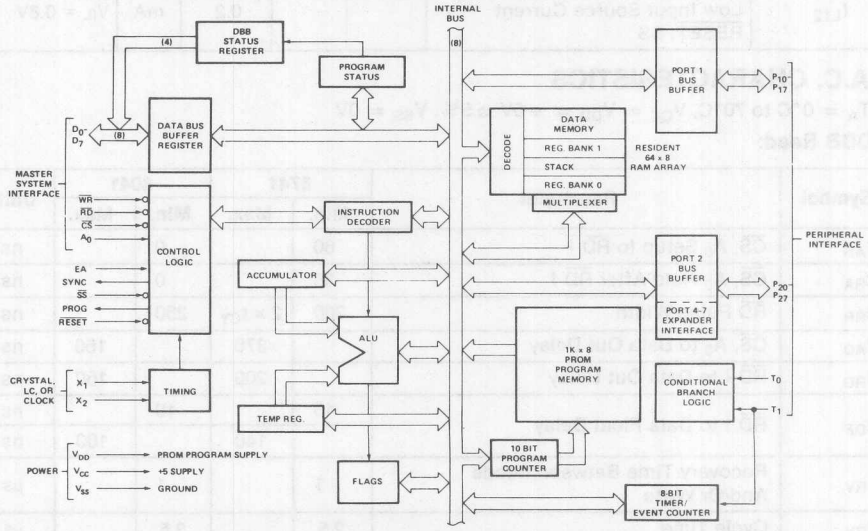
The device has two 8-bit, TTL compatible I/O ports and two test inputs. Individual port lines can function as either inputs or outputs under software control. I/O can be expanded with the 8243 device which is directly compatible and has 16 I/O lines. An 8-bit programmable timer/counter is included in the UPI device for generating timing sequences or counting external inputs. Additional UPI features include: single 5V supply, low power standby mode (in the 8041), single-step mode for debug (in the 8741), single level interrupt, and dual working register banks.

Because it's a complete microcomputer, the UPI provides more flexibility for the designer than conventional LSI interface devices. It is designed to be an efficient controller as well as an arithmetic processor. Applications include keyboard scanning, printer control, display multiplexing and similar functions which involve interfacing peripheral devices to microprocessor systems.

PIN CONFIGURATION



BLOCK DIAGRAM



© Intel Corporation, 1978

February, 1978

PRELIMINARY
Notice: This is not a final specification. Some parameter limits are subject to change.

ABSOLUTE MAXIMUM RATINGS*

Ambient Temperature Under Bias 0°C to 70°C
Storage Temperature -65°C to +150°C
Voltage on Any Pin With
Respect to Ground -0.5V to +7V
Power Dissipation 1.5 Watt

*COMMENT: Stresses above those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

D.C. AND OPERATING CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = V_{DD} = +5V ±5%, V_{SS} = 0V

Symbol	Parameter	Limits			Unit	Test Conditions
		Min.	Typ.	Max.		
V_{IL}	Input Low Voltage (All Except X_1, X_2)	-0.5		0.8	V	
V_{IH}	Input High Voltage (All Except X_1, X_2 $\overline{RESET}$)	2.0		V_{CC}	V	
V_{IH2}	Input High Voltage (X_1, $\overline{RESET}$)	3.0		V_{CC}	V	
V_{OL}	Output Low Voltage (D_0-D_7, Sync)			0.45	V	I_{OL} = 2.0 mA
V_{OL2}	Output Low Voltage (All Other Outputs Except Prog)			0.45	V	I_{OL} = 1.6 mA
V_{OH}	Output High Voltage (D_0-D_7)	2.4			V	I_{OH} = −400 μA
V_{OH1}	Output High Voltage (All Other Outputs)	2.4			V	I_{OH} = −50 μA
I_{IL}	Input Leakage Current (T_0, T_1, $\overline{RD}$, $\overline{WR}$, $\overline{CS}$, A_0, EA)			±10	μA	$V_{SS} \le V_{IN} \le V_{CC}$
I_{OL}	Output Leakage Current (D_0-D_7, High Z State)			±10	μA	V_{SS} + 0.45 ≤ $V_{IN} \le V_{CC}$
I_{DD}	V_{DD} Supply Current		10	25	mA	
$I_{CC} + I_{DD}$	Total Supply Current		65	135	mA	
V_{OL3}	Output Low Voltage (Prog)			0.45	V	I_{OL} = 1.0 mA
I_{LI1}	Low Input Source Current P_{10}-P_{17} P_{20}-P_{27}			0.4	mA	V_{IL} = 0.8V
I_{LI2}	Low Input Source Current $\overline{RESET}$, SS			0.2	mA	V_{IL} = 0.8V

A.C. CHARACTERISTICS

T_A = 0°C to 70°C, V_{CC} = V_{DD} = +5V ±5%, V_{SS} = 0V

DBB Read:

Symbol	Parameter	8741		8041		Units	Test Conditions
		Min.	Max.	Min.	Max.		
t_{AR}	$\overline{CS}$, A_0 Setup to $\overline{RD}$ ↓	60		0		ns	
t_{RA}	$\overline{CS}$, A_0 Hold After $\overline{RD}$ ↑	30		0		ns	
t_{RR}	$\overline{RD}$ Pulse Width	300	2 × t_{CY}	250		ns	t_{CY} = 2.5 μs
t_{AD}	$\overline{CS}$, A_0 to Data Out Delay		370		150	ns	
t_{RD}	$\overline{RD}$ ↓ to Data Out Delay		200		150	ns	
t_{DF}	$\overline{RD}$ ↑ to Data Float Delay	10		10		ns	
			140		100	ns	
t_{RV}	Recovery Time Between Reads And/Or Write	1		1		μs	
t_{CY}	Cycle Time	2.5		2.5		μs	6 MHz Crystal

PRELIMINARY
Notice: This is not a final specification. Some parametric limits are subject to change.

DBB Write:

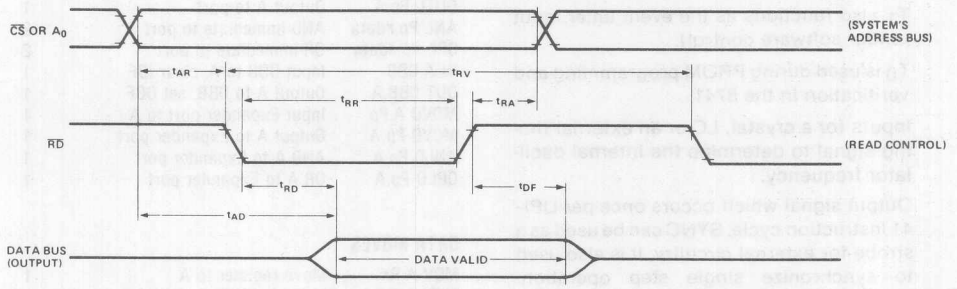
Symbol	Parameter	8741		8041		Units	Test Conditions
		Min.	Max.	Min.	Max.		
t_{AW}	$\overline{CS}$, A_0 Setup to $\overline{WR}$ ↓	60		0		ns	
t_{WA}	CS, A_0 Hold After WR ↑	30		0		ns	
t_{WW}	WR Pulse Width	300	$2 \times t_{CY}$	250		ns	$t_{CY} = 2.5\ \mu s$
t_{DW}	Data Setup to WR ↑	250		150		ns	
t_{WD}	Data Hold After WR ↑	30		0		ns	

A.C. TEST CONDITIONS

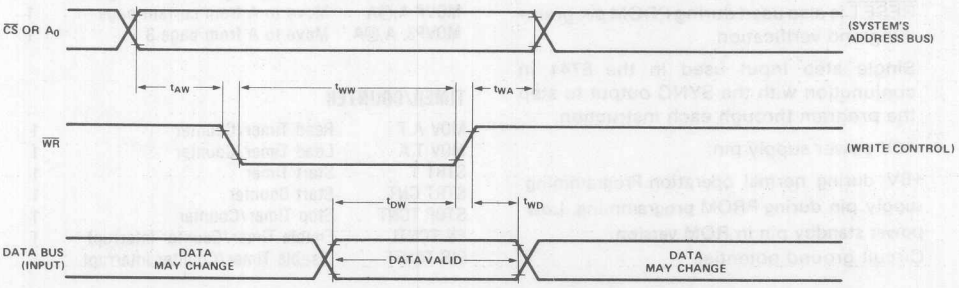
D_7-D_0 Outputs R_L = 2.2k to V_{SS}
4.3k to V_{CC}
C_L = 100 pF

WAVEFORMS

Read Operation — Data Bus Buffer Register



Write Operation — Data Bus Buffer Register



PIN DESCRIPTION

Signal	Description
D_0-D_7	Three-state, bi-directional, DATA BUS BUFFER lines used to interface the UPI-41 to an 8-bit master system data bus.
P_{10}-P_{17}	8-bit, PORT 1, quasi-bi-directional I/O lines.
P_{20}-P_{27}	8-bit, PORT 2, quasi-bi-directional I/O lines The lower 4-bits (P_{20}-P_{23}) interface directly to the 8243 I/O expander device and contain address and data information during PORT 4-7 access.
$\overline{WR}$	I/O write input which enables the master CPU to write data and command words to the UPI-41 DATA BUS BUFFER.
$\overline{RD}$	I/O read input which enables the master CPU to read data and status words from the DATA BUS BUFFER or status register.
$\overline{CS}$	Chip select input used to select one UPI-41 out of several connected to a common data bus.
A_0	Address input used by the master processor to indicate whether byte transfer is data or command.
T_0, T_1	Input pins which can be directly tested using conditional branch instructions. T_1 also functions as the event timer input (under software control). T_0 is used during PROM programming and verification in the 8741.
X_1, X_2	Inputs for a crystal, LC or an external timing signal to determine the internal oscillator frequency.
SYNC	Output signal which occurs once per UPI-41 instruction cycle. SYNC can be used as a strobe for external circuitry; it is also used to synchronize single step operation.
EA	External access input which allows emulation, testing and PROM/ROM verification.
PROG	Multifunction pin used as the program pulse input during PROM programming. During I/O expander access the PROG pin acts as an address/data strobe to the 8243.
$\overline{RESET}$	Input used to reset status flip-flops and to set the program counter to zero. $\overline{RESET}$ is also used during PROM programming and verification.
$\overline{SS}$	Single step input used in the 8741 in conjunction with the SYNC output to step the program through each instruction.
V_{CC}	+5V power supply pin.
V_{DD}	+5V during normal operation.Programming supply pin during PROM programming. Low power standby pin in ROM version.
V_{SS}	Circuit ground potential.

UPI INSTRUCTION SET

Mnemonic	Description	Bytes	Cycles
ACCUMULATOR			
ADD A,Rr	Add register to A	1	1
ADD A,@Rr	Add data memory to A	1	1
ADD A,#data	Add immediate to A	2	2
ADDC A,Rr	Add immed. to A with carry	1	1
ADDC A,@Rr	Add immed. to A with carry	1	1
ADDC A,#data	Add immed. to A with carry	2	2
ANL A,Rr	AND register to A	1	1
ANL A,@Rr	AND data memory to A	1	1
ANL A,#data	AND immediate to A	2	2
ORL A,Rr	OR register to A	1	1
ORL A,@Rr	OR data memory to A	1	1
ORL A,#data	OR immediate to A	2	2
XRL A,Rr	Exclusive OR register to A	1	1
XRL A,@Rr	Exclusive OR data memory to A	1	1
XRL A,#data	Exclusive OR immediate to A	2	2
INC A	Increment A	1	1
DEC A	Decrement A	1	1
CLR A	Clear A	1	1
CPL A	Complement A	1	1
DA A	Decimal Adjust A	1	1
SWAP A	Swap digits of A	1	1
RL A	Rotate A left	1	1
RLC A	Rotate A left through carry	1	1
RR A	Rotate A right	1	1
RRC A	Rotate A right through carry	1	1
INPUT/OUTPUT			
IN A,Pp	Input port to A	1	2
OUTL Pp,A	Output A to port	1	2
ANL Pp,#data	AND immediate to port	2	2
ORL Pp,#data	OR immediate to port	2	2
IN A,DBB	Input DBB to A, clear IBF	1	1
OUT DBB,A	Output A to DBB, set OBF	1	1
MOVD A,Pp	Input Expander port to A	1	2
MOVD Pp,A	Output A to Expander port	1	2
ANLD Pp,A	AND A to Expander port	1	2
ORLD Pp,A	OR A to Expander port	1	2
DATA MOVES			
MOV A,Rr	Move register to A	1	1
MOV A,@Rr	Move data memory to A	1	1
MOV A,#data	Move immediate to A	2	2
MOV Rr,A	Move A to register	1	1
MOV @Rr,A	Move A to data memory	1	1
MOV Rr,#data	Move immediate to register	2	2
MOV @Rr,#data	Move immediate to data memory	2	2
MOV A,PSW	Move PSW to A	1	1
MOV PSW,A	Move A to PSW	1	1
XCH A,Rr	Exchange A and register	1	1
XCH A,@Rr	Exchange A and data memory	1	1
XCHD A,@Rr	Exchange digit of A and register	1	1
MOVP A,@A	Move to A from current page	1	2
MOVP3, A,@A	Move to A from page 3	1	2
TIMER/COUNTER			
MOV A,T	Read Timer/Counter	1	1
MOV T,A	Load Timer/Counter	1	1
STRT T	Start Timer	1	1
STRT CNT	Start Counter	1	1
STOP TCNT	Stop Timer/Counter	1	1
EN TCNTI	Enable Timer/Counter Interrupt	1	1
DIS TCNTI	Disable Timer/Counter Interrupt	1	1

Mnemonic	Description	Bytes	Cycles
CONTROL			
EN I	Enable IBF Interrupt	1	1
DIS I	Disable IBF Interrupt	1	1
SEL RB0	Select register bank 0	1	1
SEL RB1	Select register bank 1	1	1
NOP	No Operation	1	1
REGISTERS			
INC Rr	Increment register	1	1
INC @Rr	Increment data memory	1	1
DEC Rr	Decrement register	1	1
SUBROUTINE			
CALL addr	Jump to subroutine	2	2
RET	Return	1	2
RETR	Return and restore status	1	2
FLAGS			
CLR C	Clear Carry	1	1
CPL C	Complement Carry	1	1
CLR F0	Clear Flag 0	1	1
CPL F0	Complement Flag 0	1	1
CLR F1	Clear F1 Flag	1	1
CPL F1	Complement F1 Flag	1	1
BRANCH			
JMP addr	Jump unconditional	2	2
JMPP @A	Jump indirect	1	2
DJNZ R,addr	Decrement register and skip	2	2
JC addr	Jump on Carry = 1	2	2
JNC addr	Jump on Carry = 0	2	2
JZ addr	Jump on A Zero	2	2
JNZ addr	Jump on A not Zero	2	2
JT0 addr	Jump on T0 = 1	2	2
JNT0 addr	Jump on T0 = 0	2	2
JT1 addr	Jump on T1 = 1	2	2
JNT1 addr	Jump on T1 = 0	2	2
JF0 addr	Jump on F0 Flag = 1	2	2
JF1 addr	Jump on F1 Flag = 1	2	2
JTF addr	Jump on Timer Flag = 1, Clear Flag	2	2
JNIBF addr	Jump on IBF Flag = 0	2	2
JOBF addr	Jump on OBF Flag = 1	2	2
JBb addr	Jump on Accumulator Bit	2	2

APPLICATIONS

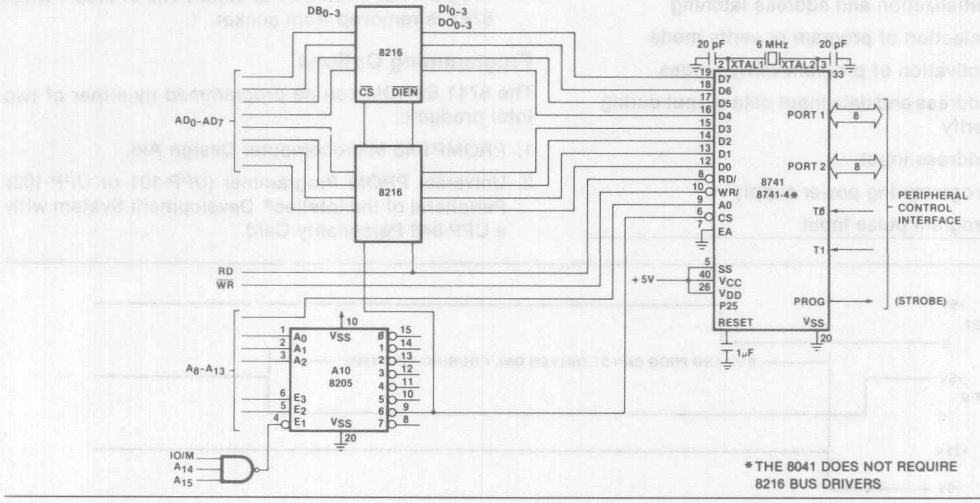


Figure 1. Recommended 8741 Interface to an 8085 System

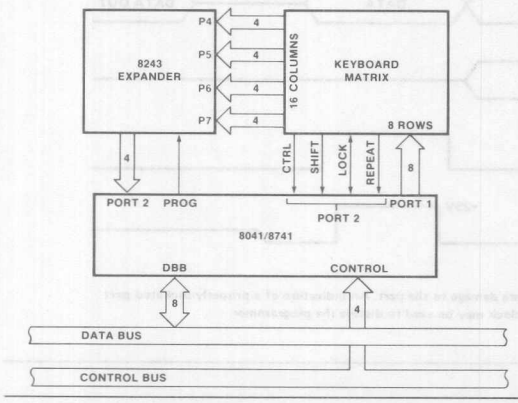


Figure 2. 8041-8243 Keyboard Scanner

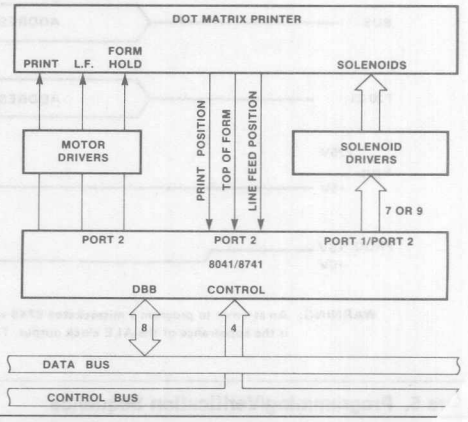


Figure 3. 8041 Matrix Printer Interface

PROGRAMMING, VERIFYING, AND ERASING THE 8741 EPROM

Programming/Verification

In brief, the programming process consists of: activating the program mode, applying an address, latching the address, applying data, and applying a programming pulse. Each word is programmed completely before moving on to the next and is followed by a verification step. The following is a list of the pins used for programming and a description of their functions:

Pin	Function
XTAL 1	Clock input (1 to 6 MHz)
RESET	Initialization and address latching
TEST 0	Selection of program or verify mode
EA	Activation of program/verify modes
BUS	Address and data input data output during verify
P20-1	Address input
V_{DD}	Programming power supply
PROG	Program pulse input

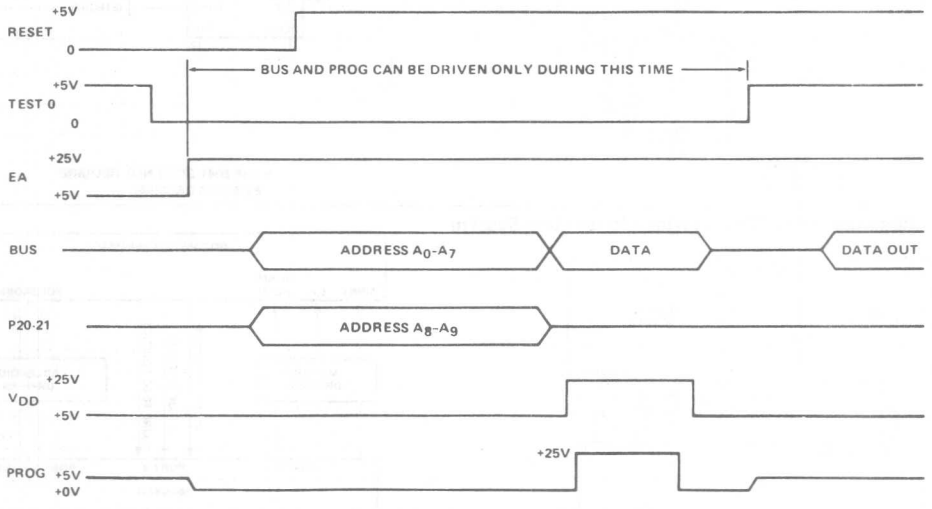
The program/verify sequence is:

1. V_{DD} = 5V, clock applied or internal oscillator operating, RESET = 0V, TEST 0 = 5V, EA = 5V, BUS and PROG floating.
2. Insert 8741 in programming socket.
3. TEST 0 = 0V (select program mode).
4. EA = 25V (activate program mode).
5. Address applied to BUS and P20-1.
6. RESET = 5V (latch address).
7. Data applied to BUS.
8. V_D = 25V (programming power).
9. PROG = 0V followed by one 50 ms pulse to 25V.
10. V_{DD} = 5V.
11. TEST 0 = 5V (verify mode).
12. Read and verify data on BUS.
13. TEST 0 = 0V.
14. RESET = 0V and repeat from step 5.
15. Programmer should be at conditions of step 1 when 8741 is removed from socket.

Programming Options

The 8741 EPROM can be programmed by either of two Intel products:

1. PROMPT-48 Microcomputer Design Aid.
2. Universal PROM Programmer (UPP-101 or UPP-102) Peripheral of the Intellec® Development System with a UPP-848 Personality Card.



WARNING: An attempt to program a missocketed 8748 will result in severe damage to the part. An indication of a properly socketed part is the appearance of the ALE clock output. The lack of this clock may be used to disable the programmer.

Figure 5. Programming/Verification Sequence

8741 Erasure Characteristics

The erasure characteristics of the 8741 are such that erasure begins to occur when exposed to light with wavelengths shorter than approximately 4000 Angstroms (Å). It should be noted that sunlight and certain types of fluorescent lamps have wavelengths in the 3000–4000Å range. Data show that constant exposure to room level fluorescent lighting could erase the typical 8748 in approximately 3 years while it would take approximately one week to cause erasure when exposed to direct sunlight. If the 8741 is to be exposed to these types of lighting conditions for extended periods of time, opaque labels are available from Intel which should be placed over the 8741 window to prevent unintentional erasure.

The recommended erasure procedure for the 8741 is exposure to shortwave ultraviolet light which has a wavelength of 2537Å. The integrated dose (i.e., UV intensity × exposure time) for erasure should be a minimum of 15 W-sec/cm^2. The erasure time with this dosage is approximately 15 to 20 minutes using an ultraviolet lamp with a 12,000 μW/cm^2 power rating. The 8741 should be placed within one inch of the lamp tubes during erasure. Some lamps have a filter on their tubes which should be removed before erasure.

A.C. TIMING SPECIFICATION FOR PROGRAMMING

T_A = 25°C ±5°C, V_{CC} = 5V ±5%, V_{DD} = 25V ±1V

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
t_{AW}	Address Setup Time to $\overline{\text{RESET}}$ ↑	4tcy			
t_{WA}	Address Hold Time After $\overline{\text{RESET}}$ ↑	4tcy			
t_{DW}	Data in Setup Time to PROG ↑	4tcy			
t_{WD}	Data in Hold Time After PROG ↓	4tcy			
t_{PH}	$\overline{\text{RESET}}$ Hold Time to Verify	4tcy			
t_{VDDW}	V_{DD}	4tcy			
t_{VDDH}	V_{DD} Hold Time After PROG ↓	0			
t_{PW}	Program Pulse Width	50	60	MS	
t_{TW}	Test 0 Setup Time for Program Mode	4tcy			
t_{WT}	Test 0 Hold Time After Program Mode	4tcy			
t_{DO}	Test 0 to Data Out Delay		4tcy		
t_{WW}	$\overline{\text{RESET}}$ Pulse Width to Latch Address	4tcy			
t_r, t_f	V_{DD} and PROG Rise and Fall Times	0.5	2.0	μs	
t_{CY}	CPU Operation Cycle Time	5.0		μs	
t_{RE}	$\overline{\text{RESET}}$ Setup Time Before EA ↑	4tcy			

Note: If TEST 0 is high, t_{DO} can be triggered by $\overline{\text{RESET}}$ 1.

D.C. SPECIFICATION FOR PROGRAMMING

T_A = 25°C ±5°C, V_{CC} = 5V ±5%, V_{DD} = 25V ±1V

Symbol	Parameter	Min.	Max.	Unit	Test Conditions
V_{DOH}	V_{DD} Program Voltage High Level	24.0	26.0	V	
V_{DDL}	V_{DD} Voltage Low Level	4.75	5.25	V	
V_{PH}	PROG Program Voltage High Level	21.5	24.5	V	
V_{PL}	PROG Voltage Low Level		0.2	V	
V_{EAH}	EA Program or Verify Voltage High Level	21.5	24.5	V	
V_{EAL}	EA Voltage Low Level		5.25	V	
I_{DD}	V_{DD} High Voltage Supply Current		30.0	mA	
I_{PROG}	PROG High Voltage Supply Current		16.0	mA	
I_{EA}	EA High Voltage Supply Current		1.0	mA	

Chapter 8
SUPPORT PRODUCTS



SUPPORT PRODUCTS

Intellec®, Prompt 48™, MCS-48™ Microcomputer Design Aid 8-1
Intellec® Microcomputer Development System .. 8-7
MCS-48™ Diskette-Based Software Support Package 8-12
MDS-48-ICE 8048 In-Circuit Emulator 8-14
8021 Emulation Board 8-19
MCS-48™ System Workshop 8-22

INTELLEC PROMPT 48
MCS-48 MICROCOMPUTER DESIGN AID

Complete low cost design aid and EPROM programmer for revolutionary MCS-48 single component computers

Simplifies microcomputing, allowing user to enter, run, debug, and save machine language programs with calculator-like ease

Utilizes two removable 8-bit MCS-48 CPUs

- **8748 CPU with erasable, reprogrammable on-chip program memory**
- **8035 CPU with off-chip program memory**

1K-byte erasable, reprogrammable on-chip (8748), expandable program memory, 1K-byte RAM in PROMPT system

64 bytes RAM on-chip, expandable register memory

256 bytes expandable RAM data memory in PROMPT system

27 on-chip TTL compatible expandable I/O lines

On-chip clock, internal timer/event counter, two vectored interrupts, eight level stack control

Single +5V DC system power requirement

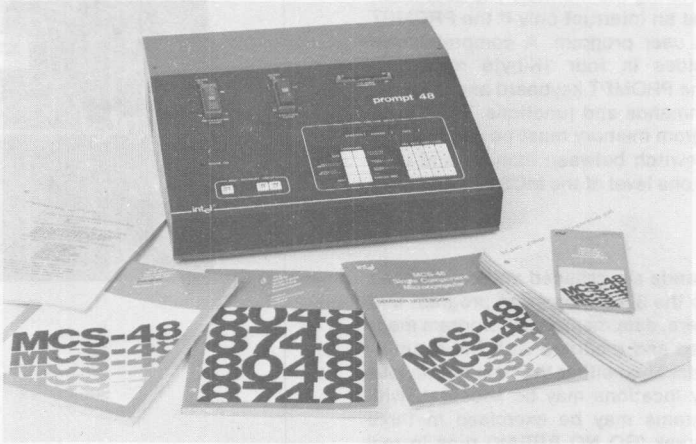
Integral keyboard and displays (no teletypewriter or CRT terminal required)

Extensive PROMPT 48 monitor, allowing system I/O, bus, and memory expansion

Compatible with Intellec microcomputer development systems

Includes comprehensive design library

The Intellec Prompt 48 MCS-48 Microcomputer Design Aid is a low cost, fully-assembled design aid for the revolutionary 8748 single component microcomputer. PROMPT 48 simplifies the programming of MCS-48 systems — programs may be entered and debugged with calculator-like ease on the large, informative display and keyboard panel. The comprehensive design library with tutorial manual is ideal for newcomers to microcomputing. PROMPT 48's panel connector allows easy access to I/O ports and system bus. Thus users can expand program memory beyond the 1K bytes provided internally. PROMPT 48 can serve as an economical 8748 Specialized PROM Programmer (SPP) peripheral in Intellec microcomputer development systems.



Intel, Intellec, INSITE, Library Manager, MCS, MEGACHASSIS, MICROMAP, PROMPT, μSCOPE, MULTIBUS, RMX/80, UPI-41, ICE, and iSBC are trademarks of Intel Corporation.

© Intel Corporation, 1978

FEATURES

Single Component Computer

The 8748 is the first microcomputer fully integrated on one component. All elements of a computing system are provided, including CPU, RAM, I/O, timer, interrupts, and erasable, reprogrammable nonvolatile program memory.

Programming Socket

PROMPT's programming socket programs this revolutionary "smart PROM" — the 8748 — in a highly reliable, convenient manner. A fail-safe interlock ensures the device is properly inserted before applying programming pulses. Each location may be individually programmed, one byte at a time. A read-before-write programming algorithm prevents device damage by inadvertently programming unerased memory.

MCS-48 Processors

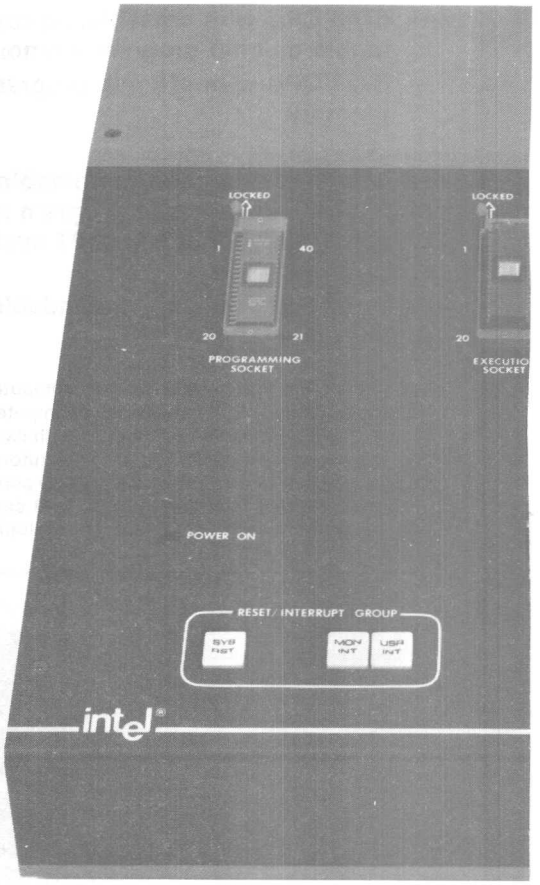
The execution socket accepts either an 8035 or an 8748 MCS-48 processor. Both are supplied with each PROMPT 48, and either can serve as heart of the PROMPT system. There are no processors within the PROMPT 48 mainframe, which instead contains monitor ROM and RAM, user RAM, peripherals, drivers, and sophisticated control circuitry. Once a processor is seated in the execution socket and power is applied, the PROMPT system comes to life. Various access modes may be selected such as program execution from PROMPT system RAM, or from on-chip PROM. Thus programs may first be executed from PROMPT RAM with the 8035 processor. When debugging is complete, the 8035 (execution socket) processor can program the 8748 (programming socket) processor. Finally, a programmed 8748 processor may be exercised by itself from the execution socket. The execution socket processor runs either monitor or user programs.

System Monitor

The system reset command initializes the PROMPT system and enters the monitor. The monitor interrupt command exits a user program gracefully, preserving system status and entering the monitor. The user interrupt command causes an interrupt only if the PROMPT system is running a user program. A comprehensive system monitor resides in four 1K-byte read only memories. It drives the PROMPT keyboard and displays and responds to commands and functions. The top 16 bytes of on-chip program memory must be used by the PROMPT system to switch between monitor and user programs. It requires one level of the MCS-48 eight-level stack.

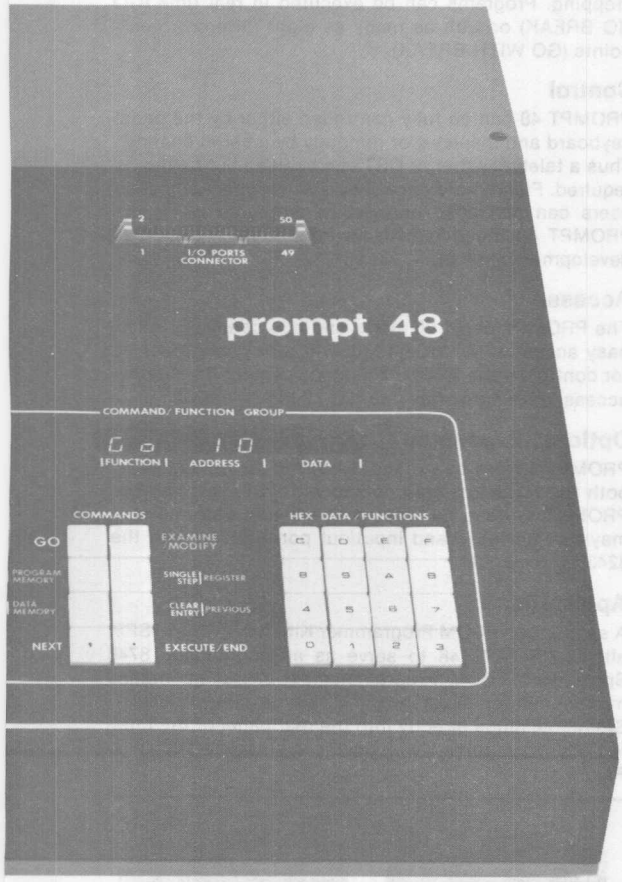
Commands

PROMPT 48's commands are grouped and color-coded to simplify access to the 8748's separate program and data memory. Registers, data memory, or program memory, may be examined and modified with the examine and modify commands. Then either the next or previous register and memory locations may be accessed with one keystroke. Programs may be exercised in three modes. The go no break (GO NO BREAK) runs in real time. The go with break (GO WITH BREAK) mode is not real time — after each instruction the MCS-48 program counter is compared against pending breakpoints. If no break is encountered, execution resumes. The go single step (GO SINGLE STEP) mode exercises one instruction at a time. Commands are like sentences, with parameters separated by ⊡ NEXT. Each command ends with ⊡ EXECUTE/END. In addition to the PROMPT basic commands, thirteen functions simplify programming. Each is started merely by pressing a hex data/function key and entering parameters as required, as shown in Table 1.



Cable Interface

An optional cable, PROMPT-SER, directly connects the PROMPT system to virtually any terminal via a rear access slot. Another cable, PROMPT-SPP, allows programs and data to be downloaded from the Intellec microcomputer development system to the PROMPT system for debugging.



Key	Function	Operation
2	Port 2 map	Allows specification of direction of each pin on port 2. Port 2 is multiplexed to address external program memory and expand I/O. Thus it must be buffered; the P2 map command establishes the direction of buffering.
3	Program EPROM	Programs 8748 EPROMs.
4	Byte search (with optional mask)	Sweeps through register, data, or program memory searching for byte matches. Starting and ending memory addresses are specified.
5	Word search (with optional mask)	Sweeps through register, data, or program memory searching for word matches. Starting and ending memory addresses are specified.
6	Hex calculator	Computes hexadecimal sums and differences.
7	8748 program for debug	Similar to program EPROM, but ensures that the top of program memory contains monitor re-entry code for debugging.
8	Compare	Verifies any portions of EPROM program memory against PROMPT memory.
9	Move memory	Allows blocks of register, data, or program memory to be moved.
A	Access	Specifies one of six access modes for PROMPT 48. For example EPROM, PROMPT RAM, or external program memory, and a variety of input/output options may be selected.
B	Breakpoint	Allows any or all of the eight breakpoints to be set and cleared.
C	Clear	Clears portions of register, data, or program memory.
D	Dump	Dumps register, data, or program memory to PROMPT's serial channel: for example, a teletypewriter paper tape punch.
E	Enter	Enters (reads) register, data, or program memory from PROMPT's serial channel.
F	Fetch	Fetches programs from EPROM to PROMPT RAM.

Table 1. PROMPT 48 Commands and Functions

Access

Easy access to the pins of the executing processor is provided via the I/O ports and bus connector. Only the EA external access, SS single step, and X1, X2 clock inputs are reserved for the PROMPT system.

Expansion

Program or data memory may be expanded beyond that provided on-chip or in the PROMPT system. I/O ports may be expanded, as with the 8243, or peripheral controllers may be memory-mapped. The I/O ports and Bus connector allows the execution socket processor to be directly interfaced to prototype systems, yet be controlled from the PROMPT panel.

Control

The command/function group panel keyboard and displays completely control PROMPT 48 — a teletypewriter or CRT terminal is not needed. A hyphen prompting character appears whenever a command or function can be entered. Addresses and data are shown whenever examining registers and memory. Parameters for commands and functions are also shown.

PROMPT stands for PROgramming Tool. It is a programmer for 8748 EPROMs, and a versatile aid for debugging MCS-48 programs. Programs can be entered via its integral panel keyboard, programming socket, or serial channel. Almost any terminal can be interfaced to the serial channel, including a teletypewriter, CRT, or an Intellec microcomputer development system. Intellec PROMPT 48 simplifies the programming of MCS-48 systems. Like the 8748 it is radically new, highly integrated, and expandable. Like the MCS-48 family, it is low cost, and ideal for small applications and programs. It is a design aid, not a development system with sophisticated software and peripherals.

MCS-48 Processors

PROMPT 48 comes complete with two of Intel's revolutionary MCS-48 processors: an 8748-4 Single Component 8-Bit Microcomputer and and 8035-4 Single Component 8-Bit Microcomputer. Advances in n-channel MOS technology allow Intel, for the first time to integrate into one 40-pin component all computer functions:

8-bit CPU
1K x 8-bit EPROM/ROM program memory
64 x 8-bit RAM data memory
27 input/output lines
8-bit timer/event counter

Performance — More than 90 instructions — each one or two cycles — make the single chip MCS-48 equal in performance to most multi-chip microprocessors. The MCS-48 is an efficient controller and arithmetic processor, with extensive bit handling, binary, and BCD arithmetic instructions. These are encoded for minimum program length; 70% are single byte operation codes, and none is more than two bytes.

Flexibility — Three interchangeable, pin-compatible devices offer flexibility and low cost in development and production, as follows:

8748 — with user-programmable and erasable EPROM program memory for prototype and pre-productions systems.

8048 — with factory-programmed mask ROM memory for low-cost, high volume production.

8035 — without program memory, for use with external program memories.

Circuitry — Each MCS-48 processor operates on a single +5V supply, with internal oscillator and clock driver, and circuitry for interrupts and resets. Extra circuitry is in the 8048 ROM processor to allow low power standby operation. The 64 x 8 RAM data memory can be independently powered.

Compatibility — For systems requiring additional compatibility, the MCS-48 can be expanded with the new 8243 I/O expander, 8155 I/O and 256-byte RAM, 8755 I/O and 2K-byte EPROM, or 8355 I/O and 2K ROM devices. MCS-48 processors readily interface to MCS-80/85 peripherals and standard memories.

Memory Capacity

PROMPT 48 is a complete, fully assembled and powered microcomputer system including program memory, data PROMPT system RAM serve as "writable program memory" — a ROM simulator for the program memory on each MCS-48 computer. 256 bytes of PROMPT system RAM serve as "external data memory," beyond the 64 register bytes on each MCS-48 computer. Users may further expand program or data memory via the panel I/O ports and bus connector.

Programming

Programs written first in assembly language, are entered in machine language and debugged with calculator-like ease on the large, informative display and keyboard panel. Most MCS-48 operations can be specified with only two keystrokes. Once entered, routines can be exercised one instruction (single step) or many instructions at a time. The principal MCS-48 register — the accumulator — is displayed while single stepping. Programs can be executed in real time (GO NO BREAK) or with as many as eight different breakpoints (GO WITH BREAK).

Control

PROMPT 48 can be fully controlled either by the panel keyboard and displays, or remotely by a serial channel. Thus a teletypewriter or CRT can be used but neither is required. Full remote control by a serial channel means users can download and debug programs using the PROMPT 48 together with an Intellec microcomputer development system.

Access

The PROMPT panel I/O ports and bus connector allow easy access to all MCS-48 pins except those reserved for control by the PROMPT system, namely EA external access, SS single step, and X1, X2 clock inputs.

Optional Expansion

PROMPT 48 may be expanded beyond the resources on both the MCS-48 single component computer and the PROMPT system. External program and data memory may be interfaced and input/out ports added with the 8243 I/O expander.

Applications

A specialized PROM Programmer Kit, the PROMPT-SPP, allows PROMPT 48 to serve as an economical 8748 Specialized PROM Programmer peripheral in Intellec microcomputer development systems. The PROMPT-SPP cable plugs directly into the rear panel of the Intellec microcomputer development system, as shown in Figure 1.

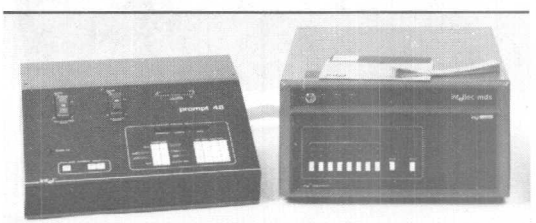


Figure 1. PROMPT 48 used as an Intellec Microcomputer Development System Peripheral

Documentation

The PROMPT 48 manual includes chapters for the reader with little or no programming experience. Topics treated range from number systems to microcomputer hardware design. A novel, unifying set of tutorial diagrams — MICROMAPS — simplify microcomputer concepts. PROMPT's handy, pocket-sized reference cardlet can be affixed to the mainframe. Programming pads aid in the organization and documentation of programs. These features, plus a comprehensive design library of manuals, articles, and application notes, make the Intellec PROMPT 48 ideal for the newcomer to microcomputing.

SPECIFICATIONS

Timing

Basic Instruction — 2.5 μs
Cycle Time — t_{CY} = 2.5 μs
Clock — 6 MHz ± 0.1%

Memory Bytes

The 8748 contains 64 bytes of register memory, no external data memory, and 1024 bytes of RAM program memory. The PROMPT system provides 256 bytes of external data memory, and 1024 bytes of RAM program memory. PROMPT RAM program memory can be used in place of the on-chip EPROM program memory; thus programs less than 1024 bytes may be designed. For larger programs additional memory can be directly interfaced to the MCS-48 bus via the PROMPT panel I/O ports and bus connector.

Memory Configuration

Memory	Maximum	On Chip	In PROMPT 48
Register	64	64	0
Data	3328	0	256
Program	4096	1024 EPROM	1024 RAM

I/O Ports

All MCS-48 I/O ports are accessible on the PROMPT panel connector.

Bus — A true bidirectional 8-bit port with associated strobes. If the bidirectional feature is not needed, bus can serve as either a statically latched output port or a non-latching input port. Input and output lines cannot be mixed.

Ports 1 and 2 — Data written to these 8-bit ports is latched and remains unchanged until written. As inputs these lines are not latching. The lines of ports 1 and 2 are called quasibidirectional. A special output structure allows each line of port 1 and half of port 2 to serve as an input, an output, or both. Any mix of input, output, and both lines is allowed.

T0, T1, and INT — Three pins that can serve as inputs. T0 can be designated as a clock output. Input/output can be expanded via the PROMPT panel connector with a special I/O expander (8243) or standard peripherals.

Reset and Interrupts

Reset — initializes the PROMPT system and enters the monitor.

Monitor Interrupt — exits a user program gracefully, preserving system status and entering the monitor.

User Interrupt — causes an interrupt only if the PROMPT system is running a user program.

The processor traps to location 3_{16}. The MCS-48 timer/event counter is not used by the PROMPT system and is available to the user. Either timer flag or interrupt will signal when overflow has occurred. The timer interrupt can be used only in the go-no-break (real time) mode.

EPROM Programming

PROMPT 48 provides a programming socket to directly program 8748s. Programs are loaded into the PROMPT RAM program memory via keyboard. EPROM, teletypewriter, or other serial interface. A fail-safe interlock ensures programming pulses are applied only if the device is properly inserted. Inadvertant reprogramming is prevented by a read-before-write programming algorithm. Each location may be individually programmed, one byte at a time.

Panel I/O Ports and Bus Connectors

All MCS-48 pins, except five, are accessible on the I/O ports and bus connector. The five reserved for PROMPT system control are EA external access, SS single step, X1, X2 crystal inputs, and 5V. Due to internal buffering of the MCS-48 bus, access times will be negligibly degraded by the PROMPT system. Since MCS-48 processors do not communicate internal address gate status, bus data must be driven out if neither PSEN nor RD is asserted.

System Devices

Both user programs and the PROMPT monitor enjoy access to system devices: serial I/O, panel displays, and keyboard. These are memory-mapped to program memory addresses beyond 2K.

Serial I/O — The serial I/O port (data 820_{16}, control 821_{16}) is defined by software and jumpers for 110 baud, 20 mA current loop, but can easily be jumpered for other baud rates and RS232C levels. Asynchronous or synchronous transmission, data format, control characters, and parity can be programmed.

Panel Displays — Eight display ports (data $810\text{-}817_{16}$) allow each of the panel displays to be written from user programs. Data written on a display device will time out after a fixed interval. Displays must be refreshed on a polled or interrupt-driven basis. User programs can call software drivers which provide this capability.

Keyboard — Software is used to debounce the panel keyboard (data 810_{16}). The monitor's input routines (see Software Drivers) provide this debouncing and can be called from user programs.

Commands

Single step / With break / No break } Go

Examine/modify { Register / Data / Program } Memory

Open previous/clear/entry ⊡ Next ⊡ Execute/End

Functions

2 Port 2 map
3 Program EPROM (8748)
4 Search (R, D or P)* memory for 1 byte, optional mask
5 Search (R, D or P) memory for 2 bytes, optional mask
6 Hexadecimal calculator +, –
7 8748 program EPROM for debug
8 Compare EPROM with memory
9 Move memory (R, D or P)
A Access
B Breakpoint
C Clear memory (R, D or P)
D Dump memory (R, D or P)
E Enter (read) memory (R, D or P)
F Fetch EPROM program memory

Note
*R, D, or P is register, data, or program.

Software Drivers

Panel Keyboard In — KBIN, KDBIN
Panel Display Out — DGS6, DGOUT, HXOUT, BLK, REFS, ENREF
Serial Channel — CI, CO, RI, PO, CSTS

Connectors

Serial I/O — 3M 3462-0001 Flat Crimp/AMP 88106-1 Flat Crimp/ TI H312113 Solder/AMP 1-583485-5 Solder.

Panel I/O Ports and Bus Connector — 3M 3425 Flat Crimp. A complete cable set including wirewrap header for prototyping is included with each PROMPT.

Equipment Supplied

PROMPT 48 mainframe with two MCS-48 processors (8748, 8035), display/keyboard, EPROM programmer, power supply, cabinet, and ROM-based monitor.
110 V AC power cable
110 or 220 V AC
Fuse
Panel I/O ports
Bus connector cable set

Physical Characteristics

Height — 5.3 in. (13.5 cm) max
Width — 17 in. (43.2 cm)
Depth — 17 in. (43.2 cm) max
Weight — 21 lb. (9.6 kg)

Electrical Characteristics

Power Requirements — either 115 or 230V AC (± 10%) may be switch selected on the mainframe. 1.8 amps max current (at 125 V AC).
Frequency — 47-63 Hz

Environmental Characteristics

Operating Temperature — 0°C to +40°C
Non-Operating Temperature — 20°C to +65°C

Reference Manuals

9800402 — Intellec PROMPT 48 User's Manual (SUPPLIED)
9800270 — MCS-48 User's Manual (SUPPLIED)
9800255 — MCS-48 and UPI-41 Assembly Language Programming Manual (SUPPLIED)

Reference manuals are shipped with each product only if designated SUPPLIED (see above). Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

ORDERING INFORMATION

Part Number	Description
PROMPT-48 or PROMPT-48-220V	Intellec PROMPT 48 MCS-48 microcomputer design aid. Complete with two MCS-48 processors (8748 and 8305), EPROM programmer, integral keyboard, displays, and system monitor in ROM.
PROMPT-SER	Serial cable for connecting PROMPT to TTY, CRT.
PROMPT-SPP	Specialized PROM programmer kit for connecting PROMPT 48 to Intellec microcomputer development systems for EPROM programming.

INTELLEC
MICROCOMPUTER DEVELOPMENT SYSTEM

Provides modular microcomputer development system for development and implementation of MCS-48, MCS-80, MCS-85, and Series 3000 microcomputer systems

Intel 8080 microprocessor, with 2 μs cycle time and 78 instructions, controls all Intellec functions

Supports assemblers for 8080, 8085, and 8748

16K bytes RAM memory expandable to 64K bytes

2K bytes ROM memory expandable with 6K or 16K PROM/ROM boards

Provides hardware interface and software drivers for TTY, CRT, line printer, high speed paper tape reader, high speed paper tape punch, and universal PROM programmer

Eight-level nested, maskable, priority interrupt system

Provides universal bus structure with multiprocessor and DMA capabilities

ROM resident system monitor includes all necessary functions for program loading, debugging and executing

RAM resident macroassembler used to assemble all 8080 machine instructions with full macro and conditional assembly capabilities

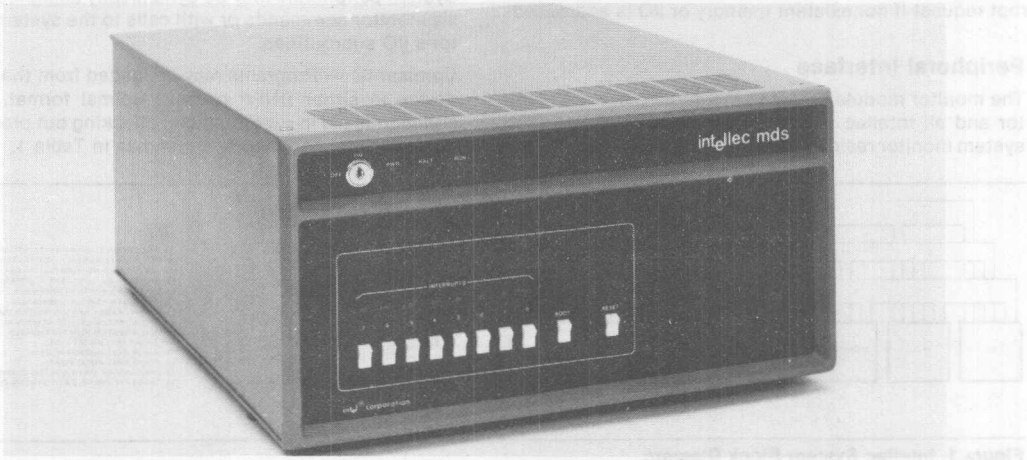
RAM resident text editor provides powerful string search, substitution, insertion, and deletion commands

ICE (in-circuit emulator) options extend Intellec diagnostic capabilities into user configured system allowing real-time emulation of user processors

Optional PROM programmer peripheral capable of programming all Intel PROMs

Optional I/O modules expandable in groups of four 8-bit input and output ports to maximum of 88 ports (all TTL compatible)

The Intellec Microcomputer Development System is a modular microcomputer development system containing all necessary hardware and software to develop and implement Intel microcomputers and microcomputer systems. The addition of peripherals and options provides the user with a complete in-circuit microcomputer development system capability, supporting product design from program development through prototype debugging, production, and field testing.



Intel, Intellec, INSITE, Library Manager, MCS, MEGACHASSIS, MICROAMP, PROMPT, μSCOPE, MULTIBUS, RMX/80, UPI-41, ICE, and iSBC are trademarks of Intel Corporation.

© Intel Corporation, 1978

FUNCTIONAL DESCRIPTION

Hardware

The standard Intellec system consists of four microcomputer modules (CPU, 16K RAM memory, front panel control, and monitor), an interconnecting printed circuit motherboard, power supplies, fans, a chassis, and a front panel. Modular expansion capability is provided by 14 additional sockets on the motherboard. A block diagram of the Intellec system is shown in Figure 1.

8080 Microprocessor

The CPU module uses Intel's powerful NMOS 8-bit 8080 microprocessor. The 8080's 2 μs cycle time, 78 instructions, unlimited subroutine nesting, vectored interrupt and DMA capabilities are fully utilized by the Intellec system. Bus control logic resolves bus contention conflicts between the CPU module and other modules capable of acquiring control of the bus. The CPU module interfaces with a sixteen line address bus and a bidirectional eight line data bus. 8080 status signals are decoded and utilized for memory and I/O operations. An eight-level, nested interrupt priority system, complete with an interrupt priority push-down stack, resolves contention for 8080 interrupt servicing.

Memory

The RAM memory module contains 16K bytes of Intel 2107A dynamic RAM operating at full processor speed. All necessary address decoding and refresh logic are contained on the module.

Control Functions

The front panel control module provides system initialization, priority arbitration, and real-time clock functions. System initialization routines reside in a 256-byte, PROM resident, bootstrap loader. An eight-level priority arbitration network resolves bus contention requests among potential bus masters. A 1 ms interrupt request generator, which can be disabled under program control, provides real-time clock functions. A 10 ms automatic time-out feature is also provided to force an interrupt request if nonexistent memory or I/O is addressed.

Peripheral Interface

The monitor module contains the Intellec system monitor and all Intellec peripheral interface hardware. The system monitor resides in a 2K-byte Intel 8316 ROM. The module contains all necessary control and data transfer circuitry to interface with the standard Intellec peripherals, including a teletype, CRT, high speed paper tape reader, high speed paper tape punch, PROM programmer, and line printer.

Bus Structure

The Intellec universal bus structure enables several CPU and DMA devices to share the bus by operating at different priority levels. Resolution of bus exchanges is synchronized by a bus clock signal which is derived independently from processor clocks. Read/write transfers may take place at rates up to 5 MHz. The bus structure contains provisions for up to 16-bit address and data transfers and is not limited to any one Intel microcomputer family.

User Interface

The Intellec front panel is intended to augment the primary user interaction medium: the system console. The simplicity of the front panel coupled with the power of the system monitor provides an efficient user/Intellec interface. The front panel contains eight interrupt request switches with corresponding indicators, CPU run and halt status indicators, a bootstrap loader switch, a reset switch, and a power on switch and indicator

Software

Resident software provided with the Intellec includes the system monitor, the 8080 macroassembler, and the text editor. Used together, these three programs simplify program preparation and speed the debugging task.

System Monitor

The system monitor provides complete control over operation of the Intellec. It is written in 8080 assembly language, resides in 2K bytes of ROM memory, and provides all necessary functions for program loading and execution. Additional commands provide extensive debug facilities and PROM programming functions. System peripherals may be dynamically assigned either via monitor commands or with calls to the system monitor's I/O subroutines.

Commands — Programs may be loaded from the reader device in either BNPF or hexadecimal format. Utility commands aid in executing and checking out programs. Typical utility commands are shown in Table 1.

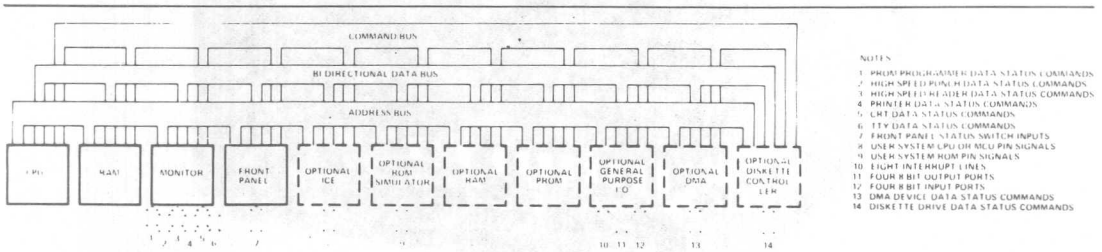


Figure 1. Intellec System Block Diagram

Command	Operation
F	Initializes memory to constant.
M	Moves a block of memory to another location.
D	Displays memory.
S	Modifies RAM memory.
X(A-F)	Examines and modifies CPU registers.
G	Sets breakpoints.
G	Initiates execution at any given address.
H	Performs hexadecimal arithmetic.
X(I)	Examines and modifies interrupt mask.

Table 1. Utility Command Functions

Input/Output — The Intellec system monitor (Intellec Series II Microcomputer Development System Monitor) contains a powerful and easily expandable input/output system built around four logical device types: a console device, a reader device, a punch device, and a list device. Associated with each logical device may be any of four physical devices. The user may control the physical device assignment to each logical device with a system command.

Peripheral Interface — Drivers are provided in the system monitor for the universal PROM programmer, ASR 33 teletype, high speed paper tape reader, high speed paper tape punch, line printer, and CRT. The user may write his own drivers for other peripheral devices and easily link them to the system monitor. All system peripherals may be accessed simply by calling I/O subroutines in the system monitor. In addition, the user may dynamically reconfigure his system by monitor commands or by calling system subroutines to assign a different physical device to each logical device. The user may also determine the current system peripheral configuration, check I/O status, and determine the size of available memory.

Macroassembler

The Intellec resident macroassembler translates symbolic 8080 assembly language instructions into the appropriate machine operation codes. It is written in PL/M-80, Intel's high level systems programming language and occupies 12K bytes of RAM memory, including space for over 800 symbols. The symbol table size may be expanded to a maximum of 6500 symbols by adding RAM memory. All I/O in the assembler is done by means of the system monitor, enabling the assembler to take advantage of the monitor's I/O system. The assembler is shipped in hexadecimal object format on paper tape or diskette and is standard with each Intellec. Full macro capability eliminates the errors of hand translation, makes it easy to modify programs by adding or deleting instructions, eliminates the need to rewrite similar sections of one code repeatedly, and simplifies program documentation.

Functions — The assembler performs its function in three passes. The first pass builds the symbol table. The second pass produces a source listing and provides error diagnostics. The third pass produces the object code. If the punch and list devices are separate (e.g., a high speed punch or printer is available) passes 2 and 3 may be combined into one pass.

Object Code — Object code produced by the assembler is in hexadecimal format. It may be loaded directly into the Intellec for execution and debugging or it may be converted by the system monitor to BNPF format for ROM programming.

Conditional Assembly — Conditional assembly permits the assembler to include or delete sections of variable code, such as the code required to handle optional external devices, which may vary from system to system.

Text Editor

The Intellec text editor is a comprehensive tool for entering and correcting assembly language programs for the Intel 8080 microcomputer. It is written in PL/M-80. It occupies 8K bytes of RAM memory, including over 4500 bytes of workspace. The workspace may be expanded to a maximum of 58K bytes by adding RAM memory. All I/O in the editor is done through the system monitor, enabling the editor to take advantage of the monitor's I/O system. The editor is shipped in hexadecimal object format on paper tape or diskette and is standard with each Intellec. The editor's command set allows the user to manipulate either entire lines of text or individual characters within a line.

Program Entry — Programs may be entered directly from the console keyboard or from the system reader device. Text is stored internally in the editor's workspace, and may be edited with the following commands:

- string insertion or deletion
- string search
- string substitution

Utility commands — To facilitate the use of these editing commands, utility commands are used to change positions in the workspace. These include:

- move pointer by line or by character
- move pointer to start of workspace
- move pointer to end of workspace

Storage — The contents of the workspace may be listed to the system console or written to the system list or punch device for future use.

OPTIONAL FEATURES

The basic Intellec capabilities may be significantly enhanced by the addition of optional features, including in-circuit emulators, the universal PROM programmer, a diskette operating system, input/output modules, RAM/PROM memory, DMA modules, and ROM simulators.

In-Circuit Emulator

In-circuit emulators (ICE) extend Intellec diagnostic capabilities into user configured systems. The Intellec resident ICE processor operates in conjunction with the host CPU and interfaces to the user system via an external cable. The ICE processor replaces the user system processor providing real-time emulation capability. Resident memory and I/O may be substituted for equivalent

user system elements, thus allowing the hardware designer to sequentially develop his system by integrating Intellec and user system hardware. Display and debug hardware eliminate the need for specially constructed user system equivalents. Augmenting these capabilities are such powerful ICE debug functions as setting breakpoints, tracing program flow, single stepping, and examining and altering CPU registers and memory locations.

Universal PROM Programmer

The Intel UPP-103 Universal PROM Programmer is an Intellec peripheral capable of programming and verifying the following Intel PROMs: 1702A, 2704, 2708, 3601, 3604, 3624, 8702A, 8704, 8708, 8748, and 8755. Programming and verification operations are initiated from the Intellec system console and are controlled by programs resident in the Intellec and universal PROM programmer.

Diskette Operating System

The addition of a single or dual drive diskette operating system significantly reduces program development time. An intelligent controller, constructed around Intel's powerful Series 3000 computing elements, provides diskette interface and control. Intel's software operating system (IDOS) in conjunction with the diskette operating system hardware provides a highly efficient and easy to use method for assembling, editing, and executing programs.

Input/Output Modules

Customized user I/O requirements may be satisfied by adding I/O modules. Each I/O module contains four 8-bit input ports (latched or unlatched), four 8-bit latched output ports (with adjustable strobe pulses), and eight system interrupt lines. All inputs and outputs are TTL compatible. Optional I/O may be expanded to a maximum of 44 input and 44 output ports.

RAM/PROM Memory

Memory may be expanded by adding RAM or PROM modules in user defined combinations. Up to 64K bytes of RAM may be added in 16K-byte increments. PROM (Intel 8702A) may be added in 256-byte increments by adding PROM modules with socket capacity for 6K bytes and populating each module with the desired mumber of PROMs. Maximum PROM capacity is 12K bytes. RAM/PROM memory overlap is resolved by giving PROM priority.

DMA Modules

DMA (direct memory access) modules work in conjunction with the Intellec MDS universal bus to maximize the efficiency of data transfers between MDS memory and selected I/O devices. Each module contains all the necessary control and data transfer logic to implement a complete DMA channel.

ROM Simulators

A ROM simulator composed of high speed bipolar RAM emulates Series 3000 bipolar microprogram ROM memory. Each ROM simulator module may be used in 512 × 16 or 1024 × 8 configurations.

SPECIFICATIONS

Word Size (Intel 8080 host processor)

Data — 8 bits
Instruction — 8, 16, or 24 bits

Memory Size

RAM — 16K bytes expandable to 64K bytes using optional modules.

ROM — 2K bytes expandable to 14K bytes in 256-byte increments using optional PROM modules.

PROM — 256 bytes expandable using optional 6K or 16K modules.

Total — RAM, ROM, and PROM may be combined in user defined configurations up to a maximum of 64K bytes.

Machine Cycle Time

Host Processor (Intel 8080) — 2.0μs

Bus Transfer Rate

Maximum bus transfer rate of 5 MHz

System Clocks

Host Processor (Intel 8080) — Crystal controlled at 2 MHz ± 0.1%
Bus Clock — Crystal controlled at 9.8304 MHz ± 0.1%

I/O Interfaces

CRT

Baud Rates: 110/300/600/1200/2400/4800/9600 (selectable)
Code Format: 7—12 level code (programmable)
Parity: Odd/even (programmable)
Interface: TTL/RS232C (selectable)

TTY

Baud Rate: 110
Input: 10 level or greater
Output: 11 level
Parity: Odd
Interface: 20 mA current loop

High Speed Paper Tape Reader

Transfer Rate: 200 cps
Control: 2-bit output, 1-bit input
Data: 8-bit byte
Interface: TTL

Punch

Transfer Rate:75 cps
Control: 2-bit output, 1-bit input
Data: 8-bit byte
Interface: TTL

Printer
Transfer Rate: 165 cps
Control: 2-bit status input, 1-bit output
Data: ASCII
Interface: TTL

PROM Programmer
Control: 3 strobes for multiplexed output data
Data: 8-bit bidirectional
Interface: TTL

General Purpose I/O (Optional)

Input Ports — 8-bit TTL compatible (latched or unlatched); expandable in 4 port increments to 44 input ports

Output Ports — 8-bit TTL compatible (latched); expandable in 4 port increments to 44

Interrupts — 8 TTL compatible interrupt lines

Interrupt

8-level, maskable, nested priority interrupt network initiated from front panel or user selected devices.

Direct Memory Access

Standard capability on Intellec bus; implemented for user selected DMA devices through optional DMA module — maximum transfer rate of 2 MHz.

Memory Access Time

RAM — 450 ns

PROM — 1.3 μs using Intel 8708A PROM

Software Capability

System Monitor — Devices supported include:
ASR 33 teletype
Intel high speed paper tape reader
Paper tape punch
CRT
Printer
Universal PROM programmer
4 recognized logical devices

Macroassembler — Accommodates 800 symbols in standard system; automatically expandable with additional RAM memory to 6500 symbols maximum; assembles all 78 8080 machine instructions plus 10 pseudo-operators.

Text Editor — 12K bytes of workspace in standard system; automatically expandable with additional RAM memory to 58K bytes.

Software Operational Environment

System Monitor — required hardware:
Intellec system
331 bytes RAM memory
2K bytes ROM memory
System console

Macroassembler — required hardware:
Intellec system
48K bytes RAM memory
System console
Reader device
Punch device
List device
Required software: system monitor

Text Editor — required hardware:
Intellec system
8K bytes RAM memory
System console
Reader device
Punch device
Required software: system monitor

Tape Format — Hexadecimal object format

Interface Cables/Connectors

MDS-920 — High speed punch interface cable
MDS-930 — Peripheral extension cable
MDS-940 — DMA cable
MDS-950 — General purpose I/O cable
MDS-960 — 25-pin connector pair
MDS-970 — 37-pin connector pair
MDS-980 — 60-pin motherboard auxiliary connector
MDS-985 — 86-pin motherboard main connector
MDS-990 — 100-pin connector hood

Emulators

MDS-ICE-30 — 3001 MCU In-Circuit Emulator
MDS-ICE-80 — 8080 In-Circuit Emulator
MDS-ICE-48 — MCS-48 8748 In-Circuit Emulator
MDS-ICE-85 — MCS-85 8085 In-Circuit Emulator

Peripherals

MDS-UPP — UPP-103 Universal PROM Programmer
MDS-PTR — High Speed Paper Tape Reader
MDS-DOS — Diskette Operating System

MCS-48 DISKETTE-BASED SOFTWARE SUPPORT PACKAGE

Extends Intellec microcomputer development system to support MCS-48 development

Takes advantage of powerful ISIS-II file handling and storage capabilities

MCS-48 assembler provides conditional assembly and macro capability

Provides assembler output in standard Intel hex format

The MCS-48 Diskette-Based Software Support Package is provided with the Intel ISIS-II system diskette and contains both the MCS-48 assembler and the diskette version of the universal PROM mapper. The assembler translates symbolic 8048 assembly language instructions into the appropriate machine operation codes, and provides both conditional and macroassembler programming. Output may be loaded either to an ICE-48 module for debugging or into an Intellec microcomputer development system for 8748 PROM programming using the universal PROM programmer.



Intel, Intellec, INSITE, Library Manager, MCS, MEGACHASSIS, MICROAMP, PROMPT, μSCOPE, MULTIBUS, RMX/80, UPI-41, ICE, and iSBC are trademarks of Intel Corporation.

© Intel Corporation, 1978

FUNCTIONAL DESCRIPTION

The MCS-48, a software support assembler package, is provided with the ISIS-II system diskette and contains both the MCS-48 assembler and the diskette version of the universal PROM mapper. The MCS-48 assembler translates symbolic 8048 assembly language instructions into the appropriate machine operation codes. The ability to refer to program addresses with symbolic names eliminates the errors of hand translation and makes it easier to modify programs when adding or deleting instructions. Conditional assembly permits the programmer to specify which portions of the master source document should be included or deleted in variations on a basic system design, such as the code required to handle optional external devices. Macro capability allows the programmer use of a single label to define a routine. The MCS-48 assembler will assemble the code required by the reserved routine whenever the macro label is inserted in the text. Output from the assembler is in standard Intel hex format. It may be either loaded directly to an in-circuit emulator (ICE-48) module for integrated hardware/software debugging, or loaded into an Intellec development system for 8748 PROM programming using the universal PROM programmer. A sample assembly listing is shown in Table 1.

```
ISIS-II 8048 MACROASSEMBLER, V1.0                                PAGE 1

LOC  OBJ                    SEQ          SOURCE STATEMENT

                              1  ;DECIMAL ADDITION ROUTINE. ADD BCD NUMBER
                              2  ;AT LOCATION 'BETA' TO BCD NUMBER AT 'ALPHA' WITH
                              3  ;RESULT IN 'ALPHA.' LENGTH OF NUMBER IS 'COUNT' DIGIT
                              4  ;PAIRS. (ASSUME BOTH BETA AND ALPHA ARE SAME LENGTH
                              5  ;AND HAVE EVEN NUMBER OF DIGITS OR MSD IS 0 IF
                              6  ;ODD)
                              7  INIT:    MACRO    AUGND,ADDND,CNT
                              8           MOV      R0, #AUGND
                              9  L1:      MOV      R1, #ADDND
                             10           MOV      R2, #CNT
                             11           ENDM
                             12  .
0001E                        13  ALPHA    EQU      30
0028                         14  BETA     EQU      40
0032                         15  COUNT    EQU      5
0100                         16           ORG      100H
                             17           INIT     ALPHA, BETA, COUNT
0100 B81E                    18+          MOV      R0, #ALPHA
0102 B928                    19+L1:       MOV      R1, #BETA
0104 BA32                    20+          MOV      R2, #COUNT
0106 97                      21           CLR      C
0107 F0                      22  LP:      MOV      A, @R0
0108 71                      23           ADDC     A, @R1
0109 57                      24           DA       A
010A A1                      25           MOV      @R0, A
010B 18                      26           INC      R0
010C 19                      27           INC      R1
010D EA07                    28           DJNZ     R2, LP
                                          END

USER SYMBOLS
ALPHA  0001E    BETA  0028    COUNT  0005    LP  0107
L1     0102

ASSEMBLY COMPLETE, NO ERRORS


ISIS-II ASSEMBLER SYMBOL CROSS REFERENCE, V1.0                   PAGE 1

SYMBOL CROSS REFERENCE

ALPHA   13#   17
BETA    14#   17
COUNT   15#   17
INIT     7#   17
L1      19#
LP      22#   28
```

Table 1. Sample MCS-48 Diskette-Based Assembly Listing

SPECIFICATIONS

Operating Environment

Required Hardware

Intellec microcomputer development system
System console
Intellec diskette operating system
32K RAM (non-macroassembler)
48K RAM (macroassembler)

Optional Hardware

Universal PROM programmer

Shipping Media

Diskette

Reference Manuals

9800255 — MCS-48/UPI-41 Assembly Language Programming Manual (SUPPLIED)

9800236 — Universal PROM Mapper Operator's Manual (SUPPLIED)

9800306 — ISIS-II User's Guide (SUPPLIED)

Reference manuals are shipped with each product only if designated SUPPLIED (see above). Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

ORDERING INFORMATION

Product Code	Description
MDS-D48	Diskette-based assembler for MCS-48 family of microprocessors.

ICE-48
MCS-48 IN-CIRCUIT EMULATOR

Emulates 8048, 8748, 8035, and 8021 Microcomputers

Extends Intellec microcomputer development system debug power to user configured system via external cable and 40-pin plug, replacing system MCS-48 device

Emulates user system MCS-48 device in real time

Shares static RAM memory with user system for program debug

Provides hardware comparators for user designated break conditions

Eliminates need for extraneous debugging tools residing in user system

Collects bus, register, and MCS-48 status information on instructions emulated

Provides capability to examine and alter MCS-48 registers, memory, and flag values, and to examine pin and port values

Integrates hardware and software efforts early to save development time

The ICE-48 MCS-48 In-Circuit Emulator module is an intellec-resident module that interfaces with any MCS-48 system. The MCS-48 family consists of the 8048, 8748, 8035, and 8021 microcomuters. The ICE-48 module interfaces with an MCS-48 system through a cable terminating in an MCS-48 pin-compatible plug which replaces the MCS-48 device in the system. With the ICE-48 plug in place, the designer has the capability to execute the system in real time while collecting up to 255 instruction cycles of real-time trace data. In addition, he can single step the system program to monitor more closely the program logic during execution. Static RAM memory is available through the ICE-48 module to emulate MCS-48 program and data memory. The designer can display and alter the contents of data and replacement RAM control memory, internal MCS-48 registers and flags, and I/O ports. Powerful debug capability is extended into the MCS-48 system while ICE-48 debug hardware and software remain inside the Intellec system. Symbolic reference capability allows the designer to use meaningful symbols rather than absolute values when examining and modifying memory, registers, flags, and I/O ports in this system.



Intel, Intellec, INSITE, Library Manager, MCS, MEGACHASSIS, MICROAMP, PROMPT, μSCOPE, MULTIBUS, RMX/80, UPI-41, ICE, and iSBC are trademarks of Intel Corporation.

© Intel Corporation, 1978

FUNCTIONAL DESCRIPTION

Debug Capability Inside User System

The ICE-48 module provides the user with the ability to debug a full prototype or production system without introducing extraneous hardware or software test tools. The module connects to the user system through the socket provided for the MCS-48 device in the user system. Intellec memory is used for the execution of the ICE-48 software. The Intellec console and file handling capabilities provide the designer with the ability to communicate with the ICE-48 module and display information on the operation of the prototype system. The ICE-48 module block diagram is shown in Figure 1.

Batch Testing

In conjunction with the ISIS-II diskette operating system, the ICE-48 module can run extensive system diagnostics without operator intervention. The designer or test engineer can define a complete diagnostic exercise, which is stored in a file on the diskette. When activated with an ISIS-II submit command, this file can instruct the ICE-48 module to execute the diagnostic routine and store the results in another file on the diskette. Results are available to the designer at his convenience. In this way, routine diagnostics and long term testing may be done without tying up valuable manpower.

Integrated Hardware/Software Development

The user prototype need consist of no more than an MCS-48 socket and timing logic to begin integration of software and hardware development efforts. Through the ICE-48 module mapping capabilities, Intellec system resources can be accessed to replace prototype memory. Hardware designs can be tested using the system software to drive the final product. Thus, the system integration phase, which can be costly when attempting to mesh completed hardware and software products, becomes a convenient two-way debug tool when begun early in the design cycle.

Real-Time Trace

The ICE-48 module captures trace information while the designer is executing programs in real time. The instructions executed, program counter, port values for port 0, port 1, and port 2, and the values of selected MCS-48 status lines are stored for the last 255 instruction cycles executed. When retrieved for display, code is disassembled for user convenience. This provides data for determining how the user system was reacting prior to emulation break, and is available whether the break was user initiated or the result of an error condition. For more detailed information on the actions of internal registers, flags, or other system operations, the user may operate in single or multiple step sequences tailored to system debug needs.

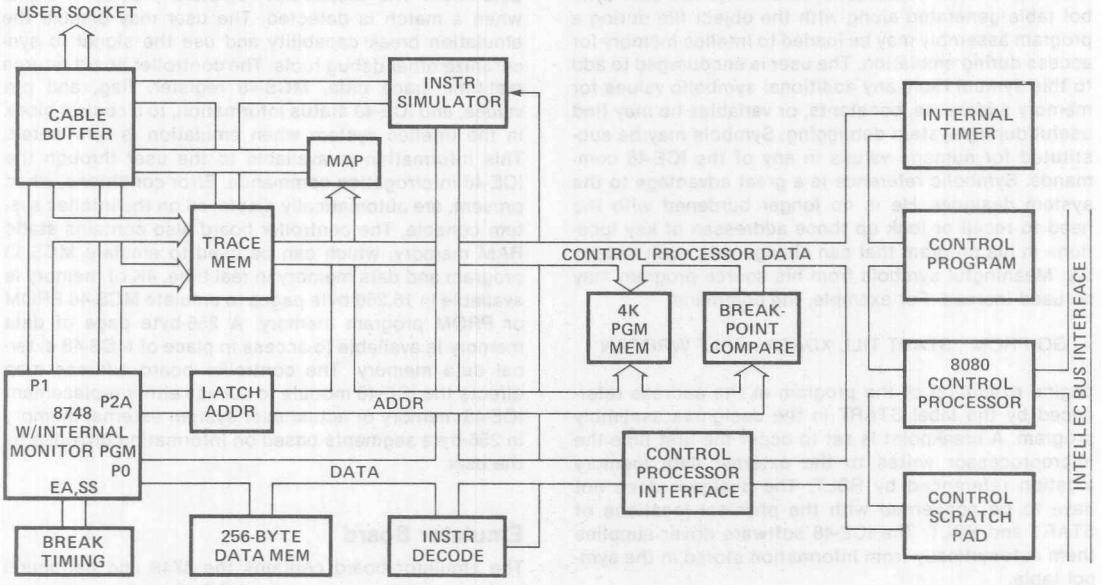


Figure 1. ICE-48 Module Block Diagram

Memory Mapping

The 8748 and 8048 contain internal program and data memory. Both program and data memory can be expanded using external memory devices.

Internal Memory — When the MCS-48 microcomputer is replaced by the ICE-48 socket in a system, the ICE-48 module supplies static RAM memory as a replacement for the internal microcomputer memory. The ICE-48 module has enough RAM memory available to emulate up to the total 4K control memory capability of the system. The ICE-48 module also provides for up to 320 bytes of data memory.

External Memory — The ICE-48 module separates replacement control memory into 16 256-byte blocks. Replacement external data memory consists of one 256-byte block. Each block of memory can be defined separately as supplied by the user system or supplied by the ICE-48 module. The user may assign ICE-48 equivalent memory to take the place of external memory not yet supplied in his system. During final debug stages when external PROM or resident 8748 is used for program execution, the designer may load the program back to ICE-48 memory to test out program changes before reassembly and reprogramming the PROM.

Symbolic Debugging

ICE-48 software provides symbolic definition of all MCS-48 registers, flags, and selected MCS-48 pins. Symbolically defined pseudo registers provide access to the sense of MCS-48 flip flops which enable time, counter, interrupt, and flag-0/flag-1 options. In addition, the user may reference locations in program and data memory, or their contents, symbolically. The user symbol table generated along with the object file during a program assembly may be loaded to Intellec memory for access during emulation. The user is encouraged to add to this symbol table any additional symbolic values for memory addresses, constants, or variables he may find useful during system debugging. Symbols may be substituted for numeric values in any of the ICE-48 commands. Symbolic reference is a great advantage to the system designer. He is no longer burdened with the need to recall or look up those addresses of key locations in his program that can change with each assembly. Meaningful symbols from his source program may be used instead. For example, the command:

```
GO FROM . START TILL XDATA . RSLT WRITTEN
```

begins execution of the program at the address referenced by the label START in the designers assembly program. A breakpoint is set to occur the first time the microprocessor writes to the external data memory location referenced by RSLT. The designer does not have to be concerned with the physical locations of START and RSLT. The ICE-48 software driver supplies them automatically from information stored in the symbol table.

Hardware

The ICE-48 module is a microcomputer system utilizing Intel's 8748 microcomputer as its nucleus. The 8748 provides the MCS-48 emulation characteristics. The ICE-48 module uses an Intel 8080 to communicate with the Intellec host processor via a DMA port. The 8080 also controls an internal ICE-48 bus for intramodule communication. ICE-48 hardware consists of two PC boards, the controller board, and the emulator board, all of which reside in the Intellec chassis. A cable interfaces the ICE-48 boards to the MCS-48 system. The cable terminates in a MCS-48 pin compatible plug which replaces any MCS-48 device in the user system. The ICE-48 module block diagram is shown in Figure 1.

Real-Time Trace

Trace Buffer

While the ICE-48 module is executing the user program, it is monitoring port, program counter, data, and status lines. Values for each instruction cycle executed are stored in a 255 × 44 real-time RAM trace buffer. A resettable timer resident on the controller board counts instruction cycles and provides timing for the trace monitor.

Controller Board

The ICE-48 module talks to the Intellec system as a peripheral device. The controller board receives commands from the Intellec system and responds through a DMA port. Three 15-bit hardware breakpoint registers are available for loading by the user. While in emulation mode, a hardware comparator is constantly monitoring address and status lines for a match to terminate an emulation. The breakpoint registers provide a signal when a match is detected. The user may disable the emulation break capability and use the signal to synchronize other debug tools. The controller board returns real-time trace data, MCS-48 register, flag, and pin values, and ICE-48 status information, to a control block in the Intellec system when emulation is terminated. This information is available to the user through the ICE-48 interrogation commands. Error conditions, when present, are automatically displayed on the Intellec system console. The controller board also contains static RAM memory, which can be used to emulate MCS-48 program and data memory in real-time. 4K of memory is available in 16,256-byte pages to emulate MCS-48 PROM or PROM program memory. A 256-byte page of data memory is available to access in place of MCS-48 external data memory. The controller board address map directs the ICE-48 module to access either replacement ICE-48 memory or actual user system external memory in 256-byte segments based on information provided by the user.

Emulator Board

The emulator board contains the 8748 and peripheral logic required to emulate the MCS-48 device in the user system. A software selectable 6 MHz or 3 MHz clock drives the emulated MCS-48 device. This clock can be disabled and replaced with a user supplied TTL clock in the user system.

Cable Card

The cable card is included for cable driving. It transmits address and data bus information to the user system through a 40-pin connector which plugs into the user system in the socket designed for the MCS-48 device.

Software

The ICE-48 software driver is a RAM-based program which provides the user with an easy to use command language (see Table 1, Table 2, and Table 3) for defining breakpoints, initiating real-time emulation or single step operation, and interrogating and altering user system status recorded during emulation. The ICE-48 command language contains a broad range of modifiers to provide the user with maximum flexibility in defining the operation to be performed. The ICE-48 software driver is available on diskette and operates in 32K of Intellec RAM memory.

Command	Operation
Enable	Activates breakpoint and display registers for use with go and step commands.
Go	Initiates real-time emulation and allows user to specify breakpoints and data retrieval.
Step	Initiates emulation in single instruction increments. Each step is followed by register dump. User may optionally tailor other diagnostic activity to his needs.
Interrupt	Emulates user system interrupt.

Table 1. ICE-48 Emulation Commands

Command	Operation
Display	Prints contents of memory, MCS-48 device registers, I/O ports, flags, pins, real-time trace data, symbol table, or other diagnostic data on list device.
Change	Alters contents of memory, register, output port, or flag. Sets or alters breakpoints and display registers.
Map	Defines memory status.
Base	Establishes mode of display for output data.
Suffix	Establishes mode of display input data.

Table 2. ICE-48 Interrogation Commands

Command	Operation
Load	Fetches user symbol table and object code from input device.
Save	Sends user symbol table and object code to output device.
Define	Enters symbol name and value to user symbol table.
Move	Moves block of memory data to another area of memory.
List	Defines list device.
Exit	Returns program control to ISIS-II.
Evaluate	Converts expression to equivalent values in binary, octal, decimal, and hex.
Remove	Deletes symbols from symbol table.
Reset	Reinitializes ICE-48 program variables.

Table 3. ICE-48 Utility Commands

SPECIFICATIONS

ICE-48 Operating Environment

Required Hardware

Intellec microcomputer development system
System console
Intellec diskette operating system
ICE-48 Module

Required Software

System monitor
ISIS-II

Equipment Supplied

Printed circuit boards
Interface cables and buffer module
ICE-48 software, diskette-based version

System Clock

Crystal controlled 6.0 MHz internal, 3.0 MHz internal or user supplied TTL external: software selectable.

Physical Characteristics

Width — 12.00 in. (30.48 cm)
Height — 6.75 in. (17.15 cm)
Depth — 0.50 in. (1.27 cm)
Weight — 8.00 lb. (3.64 kg)

Electrical Characteristics

DC Power Requirements

V_{CC} = ±5V ±5%
I_{CC} = 10A max; 7.0A typ
V_{DD} = +12V ±5%
I_{DD} = 79 mA max; 45 mA typ
V_{BB} = −10V
I_{BB} = 20 mA

Input Impedance — @ICE-48 user socket pins:

V_{IL} = 0.8V (max), I_{IL} = 1.6 mA,
V_{IH} = 2.0V (min), I_{IH} = 40 μA

For Bus:
V_{IL} = 0.8V (max), I_{IL} = 250 μA
V_{IH} = 2.0V (max), I_{IH} = 20 μA

Output Impedance — @ICE-48 user socket pins:

P1, P2:
V_{OL} = 0.5V (max), I_{OL} = 16 mA
V_{OH} = V_{CC} (10K pullup)

For Bus:
V_{OL} = 0.5V (max), I_{OL} = 25 mA
V_{OH} = 3.65V (min), I_{OH} = 1 mA

Others:
V_{OL} = 0.5V (max), I_{OL} = 16 mA
V_{OH} = 2.4V (max), I_{OH} = 400 μA

Environmental Characteristics

Operating Temperature — 0°C to 40°C

Operating Humidity — Up to 95% relative humidity without condensation

Reference Manuals

9800464 — ICE-48 Operator's Manual (SUPPLIED)

Reference manuals are shipped with each product only if designated SUPPLIED (see above). Manuals may be ordered from any Intel sales representative, distributor office or from Intel Literature Department, 3065 Bowers Avenue, Santa Clara, California 95051.

ORDERING INFORMATION

Part Number	Description
MDS-48-ICE	8048, 8748, 8035, 8021 CPU in-circuit emulator. Cable assembly and interactive diskette software included.

MDS-EM1
8021 EMULATION BOARD

EPROM functional equivalent of 8021 — single component 8-bit microcomputer

Based on 8748 — user programmable/erasable EPROM 8-bit computer

Operates with ICE-48™ to provide full in-circuit debugging of 8021 prototype system

Connects to prototype system through 8021 pin compatible plug

On-card 3.0 MHz or external TTL driven clock

Portable 4" X 7" microcomputer circuit assembly

The MDS-EM1 emulator board is a ready-to-use 4" X 7" microcomputer circuit assembly that emulates the Intel 8021 microcomputer. A 12-inch flat-cable assembly connects the board to the 8021 socket in a prototype system. The board is designed so that it can be mounted either as a stand-alone unit, or within the prototype assembly.

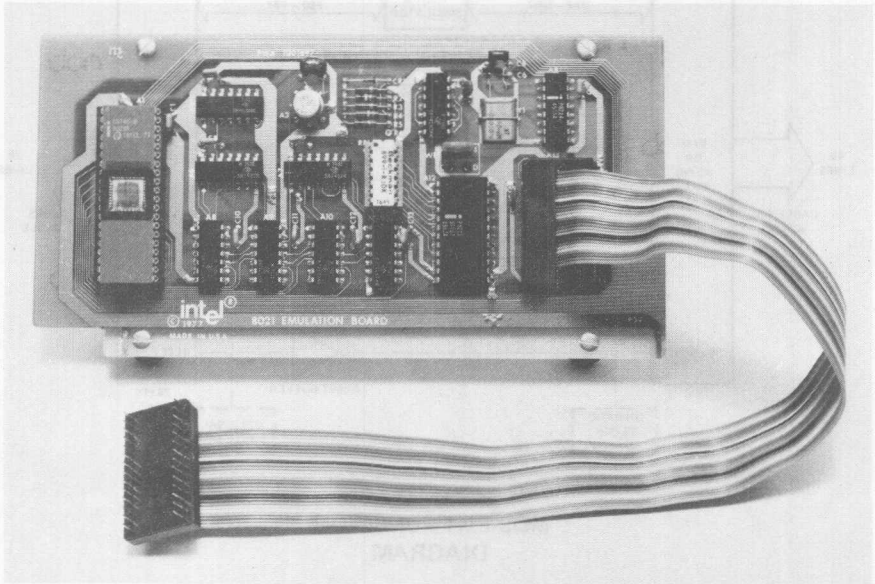
The 8021 microcomputer has 1K X 8 mask-programmable ROM program memory and 64 by 8 RAM data memory. The MDS-EM1 is controlled by an Intel 8748, with 1K of EPROM program memory and a 64 byte data memory. The EPROM can be programmed and erased repeatedly during hardware and software development. The MDS-EM1 has several ancillary circuits that perform the following functions which are specific to the 8021:

Zero crossing detector

Crystal controlled clock/buffer

Port Ø simulator

For prototype debugging, the 8748 can be removed from its socket and replaced with a cable to an INTEL ICE-48. When used with the MDS-EM1, ICE-48 emulates the 8021 in real-time, or single-steps the 8021 program at the user's command. A full range of capabilities for examining and modifying 8021 memory and status are supplied through ICE-48.



ARDWARE

The MDS-EM1 emulation board uses the 8748 to perform the emulation.

PØ SIMULATOR

Port Ø of the 8021 is a quasi*-bidirectional port. The PØ simulator converts the data bus of the 8748 into a quasi-bidirectional port.

CRYSTAL CONTROL CLOCK BUFFER

The MDS-EM1 allows user to select an on-board oscillator or a TTL clock driven from the 8021 user's prototype system via a Cambion Suitcase jumper.

Jumper	Position	State
W1	A – B	On-Board
	C – D	External TTL Clock

*A bidirectional port which serves as an input port, output port, or both even though outputs are statically latched.

ZERO CROSS DETECTION SIMULATOR

The zero cross detection simulator enables the 8748's T1 input to detect zero-crossings. The circuitry provides a high level signal on a positive crossing and a low level signal on a negative crossing of zero to the T1 input of the 8748.

RESET BUFFER

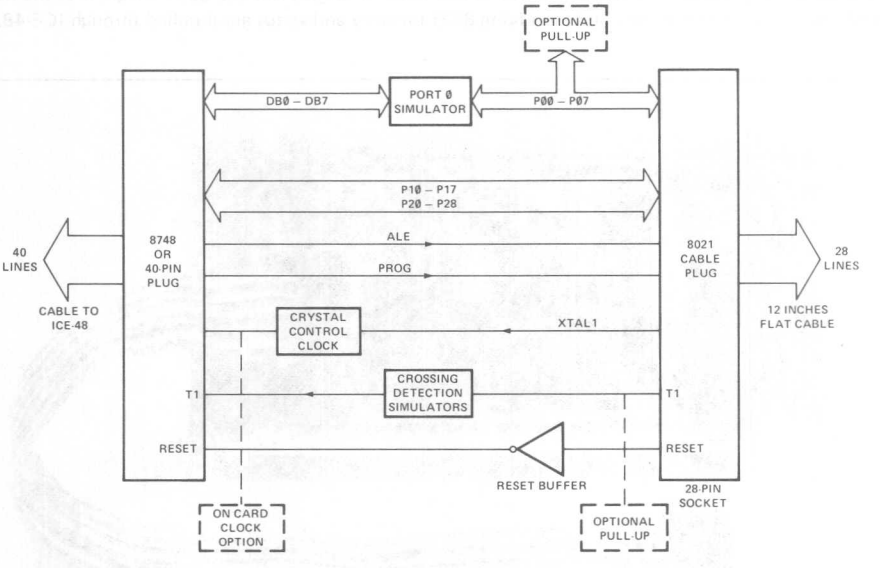
The 8021 resets on a logic HIGH level signal. However, the 8748 resets on a logic LOW level, thus an inverter is provided on the MDS-EM1 to make the two chips compatible.

OPTIONAL PULL-UPS

Resistors are provided to simulate the optional pull-up resistors on T1 input and Port Ø of the 8021. A removable resistor pack is used on Port Ø. The T1 input pull up can be installed by soldering in a 50K resistor.

SOFTWARE

When emulating the 8021 with MDS-EM1, the user must observe the 8021 instruction set.



MDS-EM1 FUNCTION DIAGRAM

SPECIFICATIONS

OPERATING ENVIRONMENT

Stand-Alone

Required Hardware:

MDS-EM1 emulation board

In-Circuit Emulation

Required Hardware:

MDS-EM1 emulation board

Intellec Microcomputer Development System configurated to support ICE-48

EQUIPMENT SUPPLIED

MDS-EM1 printed circuit board

12" long flat cable terminating in 28-pin plug, pin compatible with 8021

MDS-EM1 Operator's Manual

SYSTEM CLOCK

Crystal controlled 3.0 MHz on board or user supplied TTL external clock: hardware jumper selectable.

PHYSICAL CHARACTERISTICS

Width:	7.0 in. (17.78 cm)
Height:	4.0 in. (10.16 cm)
Depth:	0.75 in. (1.91 cm)
Weight:	<1.0 lbs. (0.45 kg)

ELECTRICAL CHARACTERISTICS

DC Power:

V_{CC} 5V ± 5%

I_{CC} 300 mA (max.)

ENVIRONMENTAL CHARACTERISTICS

Operating Temperature: 0 — 55°C

Operating Humidity: up to 95% relative humidity without condensation

ORDERING INFORMATION

Part Number	Description
MDS-EM1	8021 Emulation Board

MCS-48™ SYSTEM WORKSHOP

ourses presented at training centers and customer facilities.

Training Centers
- Boston
- Chicago
- Santa Clara

Scheduled on a continuing basis throughout the year.

System demonstrations

On-site courses tuned to customer requirements.

Hands-on laboratory sessions reinforce lecture.

Training center classes limited to 12 attendees.

REGISTRATION AND ADDITIONAL INFORMATION: Contact MCD Training at Intel Corporation, Santa Clara, California 95051, (408) 246-7501, or your local Intel sales office.

This workshop will prepare the student to design and develop a system using the Intel 8048 microprocessor through the use of lecture, demonstration and laboratory "hands-on" experience with the Intellec® Development System and PROMPT-48.

COURSE OUTLINE:

Day 1
Orientation
Introduction
a. Microprocessor System
 1. Function
 2. Organization
 3. Programming
b. 8048 Overview
 1. Functional Sections
 2. Programming Model
 3. Execution Sequence

Assembly Language Instructions
a. I/O Instructions
b. Data Move Instructions
c. Increment/Decrement Instructions
d. Branch Instructions
e. Worksession No. 1
f. Accumulator Group Instructions
 1. ADD/ADDC
 2. Logicals

PROMPT-48
a. Function
b. Operation

Laboratory Exercise
a. Program Entry and Execution using PROMPT-48

Day 2
Assembly Language Instructions
a. Accumulator Group Instructions
 1. Flags
 2. Rotates
b. Specials (XCH, DA, SWAP)
c. Worksession No. 2
d. Subroutines
 1. Invocation
 2. Stack Operation
e. Interrupt System
 1. Description
 2. Service Subroutines
 3. Multiple Source Systems

Development System
a. Function
b. Disk Operating System

Text Editor and Macro Assembler
a. Function
b. Operation

Laboratory Exercise
a. Bootstrap Procedures
b. Create, Edit and Assemble Source Program
c. Execute Program

Day 3
System Timing
a. Basic Timing and Timer
b. Bus Timing for Peripheral Devices

Peripherals and Design
a. Expanding Memory*
 1. Program Memory (1, 2K ROMs)
 2. Data Memory (RAMs)
b. Expanding Ports (8243)*
 1. Device Characteristics
 2. Software Control of Ports
c. Combination Chips*
 1. 8155 RAM and I/O Chip
 2. 8355,8755 ROM and I/O Chip
d. Peripheral Interfacing (Parallel)*
 1. 8255 Parallel I/O
 2. 8279 Keyboard and Display Interface
 —Keyboard Scanning Techniques
 —Display Refresh

Laboratory Exercise
a. Edit and Assemble Using DOS
b. Execute Using PROMPT-48

Day 4
Peripherals and Design
a. Peripheral Interfacing (Serial)*
 1. Transmission Formats
 2. Asynchronous Operation
 3. RS232C Interface
b. A/D and D/A Interfacing*
 1. Successive Approximation A/D
 2. A/D, D/A Chips
 3. A/D Design

Laboratory Exercises
a. Edit and Assemble Programs
b. Execute Programs

*Each section will consist of a design example including schematic, bus loading calculations, software and timing.

PROGRAMMABLE PERIPHERAL CIRCUITS WORKSHOP

This course will cover the Programmable Peripheral Circuits that are used in a wide variety of application areas such as process control, terminals, communications, numerical control, instrumentation, etc.

Each device is covered in sufficient depth to allow the attendee to define its hardware and software characteristics and evaluate its application areas.

COURSE OUTLINE:

Day 1
Introduction
a. Programmable Concept
 1. Initialization Commands
 2. Operation Commands
b. Addressing Methods
 1. Chip Selection
 2. Memory Mapping
 3. I/O Mapping

8253 Programmable Interval Timer

8257 Programmable DMA Controller

8259 Programmable Interrupt Controller

8279 Programmable Keyboard/Display Interface
a. Chip Descriptions and Applications
b. Programming Requirements
c. Design Examples

Day 2
8271 Programmable Floppy Disc Controller

8273 SDLC Protocol Controller

8275 Programmable CRT Controller

8155/8355/8755 Combination Memory and I/O Ports
a. Chip Descriptions and Applications
b. Programming Examples
c. Design Examples

Appendices

PACKAGING INFORMATION AND ORDERING INFORMATION



APPENDICES

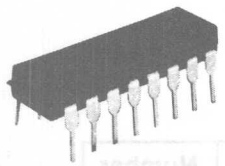
Packaging Information A1-1
Ordering Information A2-1

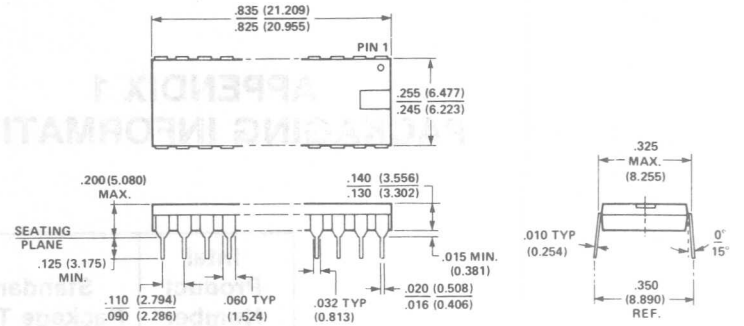
APPENDIX 1
PACKAGING INFORMATION

	Intel Product Number	Standard Package Type			Number of Pins
Microcomputers	8021			P	28
	8048		D	P	40
	8049		D	P	40
	8748	C			40
	8035		D	P	40
	8039		D	P	40
Memory and I/O Expanders	8355		D	P	40
	8755A	C			40
	8155/8156		D	P	40
I/O Expander	8243		D	P	24
Standard ROMs	8308		D	P	24
	2316E	C	D	P	24
Standard EPROM	8708	B			24
	2716	B			24
Standard RAMs	8111A-4	C	D	P	18
	8101A-4	B C		P	22
	5101	B		P	24
Standard I/O	8212		D	P	24
	8255	C		P	40
	8251	C	D	P	28
Standard Peripherals	8205		D	P	16
	8214		D	P	24
	8216		D	P	16
	8226		D	P	16
	8253	C			24
	8259		D	P	28
	8278		D	P	40
	8279		D		40
Universal Peripheral Interface	8041		D	P	40
	8741	C			40

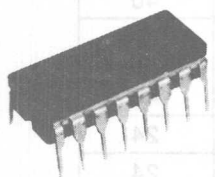
B = Black Ceramic C = Ceramic D = Ceramic DIP P = Plastic

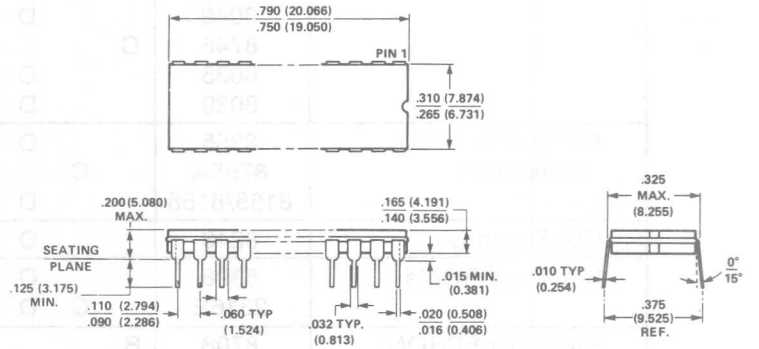
16-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P



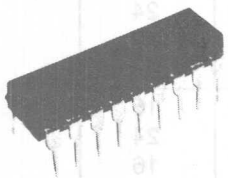


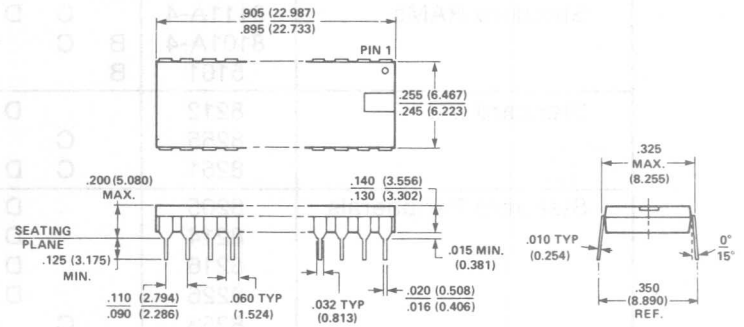
16-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



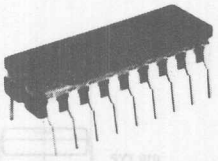


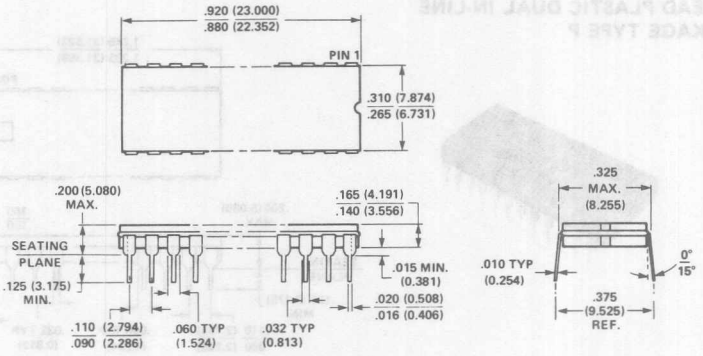
18-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P



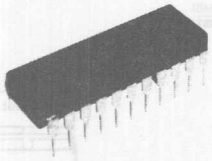


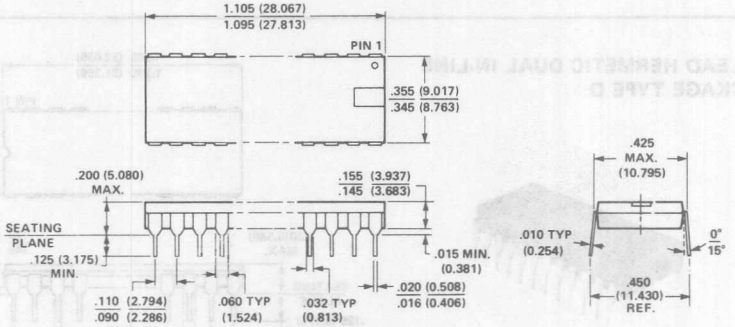
18-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



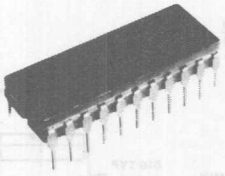


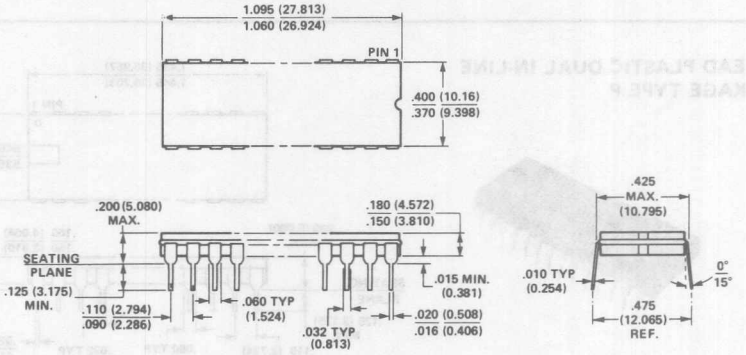
22-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P

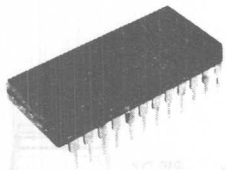


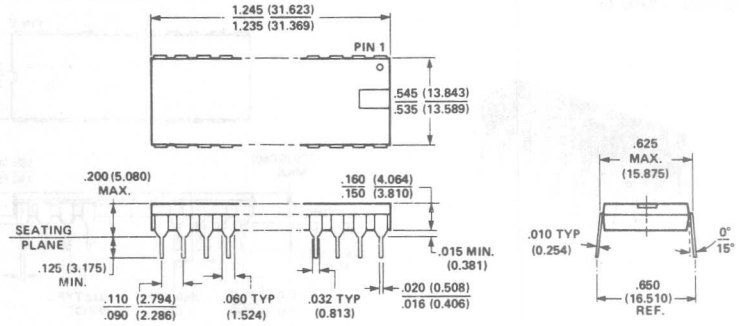


22-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D

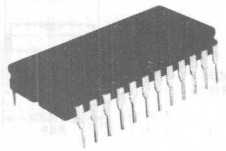


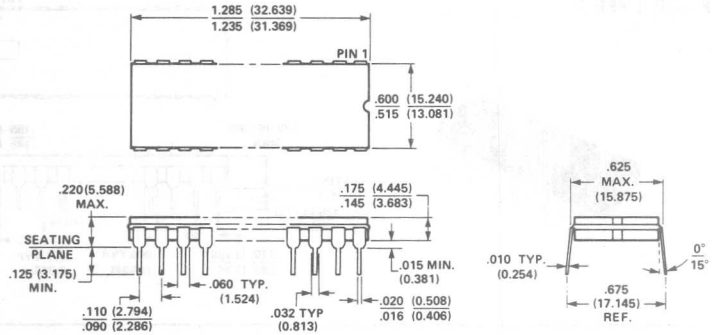




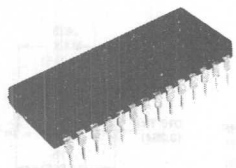


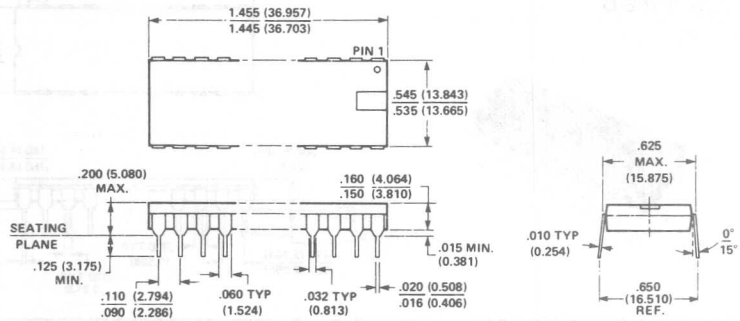
24-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



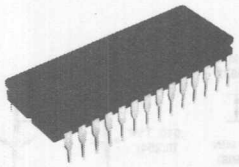


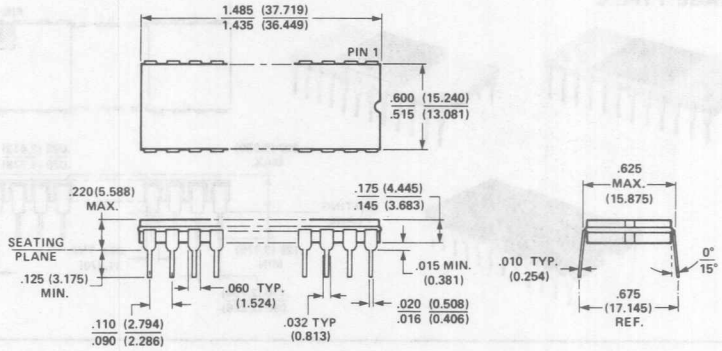
28-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P





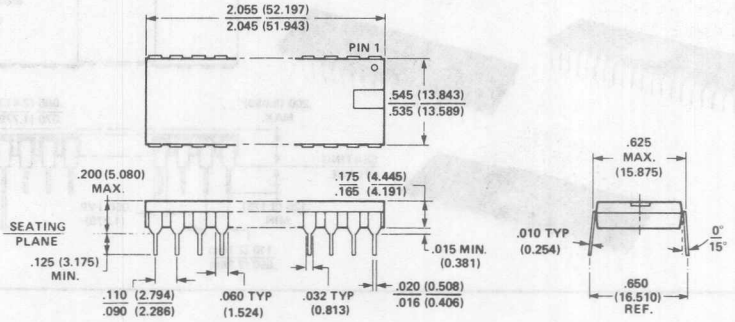
28-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



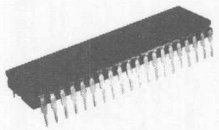


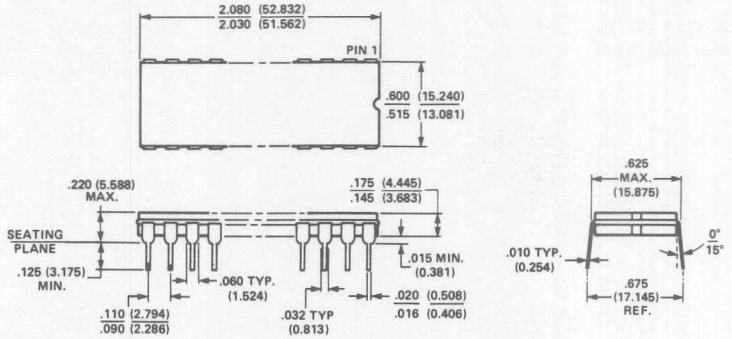
40-LEAD PLASTIC DUAL IN-LINE PACKAGE TYPE P



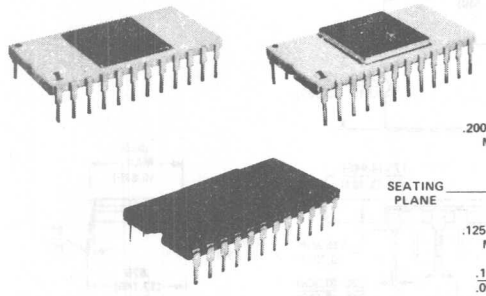


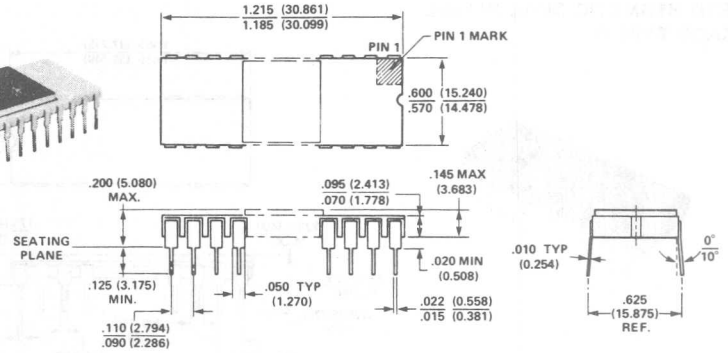
40-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE D



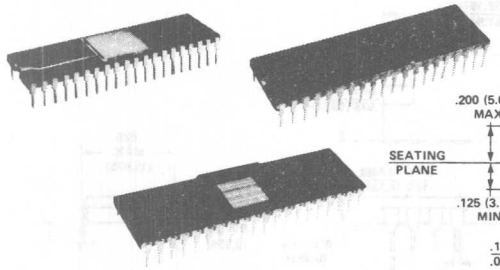


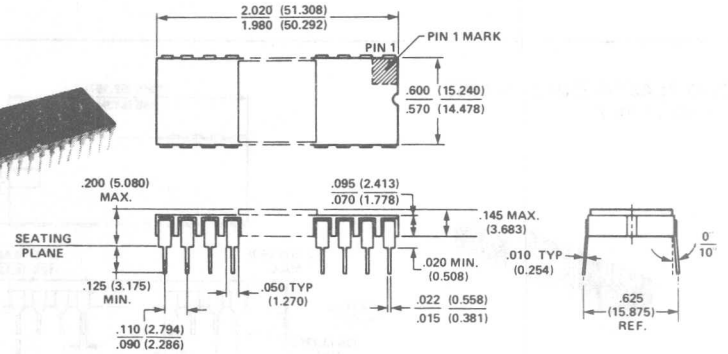
24-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C





40-LEAD HERMETIC DUAL IN-LINE PACKAGE TYPE C

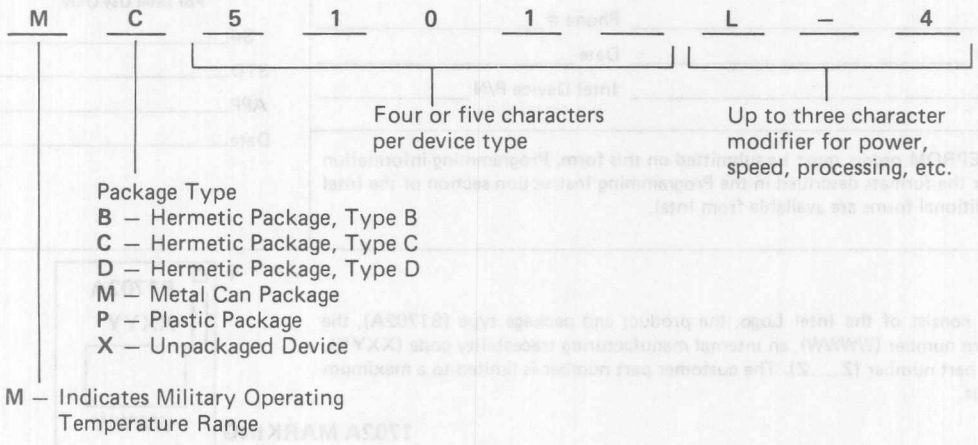




ORDERING INFORMATION

Semiconductor components are identified as follows:

Example:



Examples:

P5101L CMOS 256 X 4 RAM, low power selection, plastic package, commercial temperature range.

C8080A2 8080A Microprocessor with 1.5 μs cycle time, hermetic package Type C, commercial temperature range.

MD3604/C 512 X 8 PROM, hermetic package Type D, military temperature range, MIL-STD-883 Level C processing.*

MC8080A/B 8080A Microprocessor, hermetic package Type C, military temperature range, MIL-STD-883 Level B processing.*

Kits, boards and systems may be ordered using the part number designations in this catalog.

The latest Intel OEM price book should be consulted for availability of various options. These may be obtained from your local Intel representative or by writing directly to Intel Corporation, 3065 Bowers Avenue, Santa Clara, California 95051.

**On military temperature devices, B suffix indicates MIL-STD-883 Level B processing. Suffix C indicates MIL-STD-883 Level C processing. "S" number suffixes must be specified when entering any order for military temperature devices. All orders requesting source inspection will be rejected by Intel.*

CUSTOMER EPROM ORDER FORM A

Company _____ Phone # _____

Company Contact _____ Date _____

P.O. # _____ Intel Device P/N _____

All custom MOS EPROM orders must be submitted on this form. Programming information should be sent per the formats described in the Programming Instruction section of the Intel Data Catalog. Additional forms are available from Intel.

For Intel Use Only

S# _____

STD _____

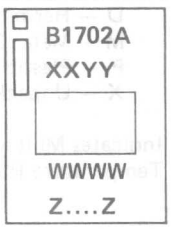
APP _____

Date _____

MARKING

The marking will consist of the Intel Logo, the product and package type (B1702A), the 4-digit Intel pattern number (WWWW), an internal manufacturing traceability code (XXYY), and the customer part number (Z....Z). The customer part number is limited to a maximum of 9 digits or spaces.

1702A MARKING EXAMPLE



FLOPPY DISK

Programming information may be sent on Intel Microcomputer Development System Floppy Disk. When using this input medium the floppy disk file name should be indicated in the Customer Part Number Section below. The type of floppy disk sent should also be indicated by checking one of the appropriate boxes:

☐ Single Density ☐ Double Density

CUSTOMER PART NUMBER

	Customer P/N (Please Fill-In)	Floppy Disk File Name (Please Fill-In)	Intel Pattern Number (Please Do Not Use)
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			
11.			
12.			
13.			
14.			
15.			
16.			
17.			
18.			
19.			

CUSTOMER 8041, 8048 ROM ORDER FORM C

Company _____ Phone # _____

Company Contact _____ Date _____

P.O. # _____ Package Type: ☐ Plastic ☐ Cerdip

For Intel Use Only

S# _____

STD _____

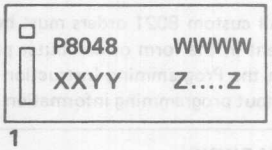
APP _____

Date _____

All custom 8041 and 8048 orders be submitted on this form. Programming information should be sent per the formats described in the Programming Instruction section of the Intel Data Catalog. Additional forms are available from Intel.

MARKING

All devices will be marked as shown at the right figure. The marking will consist of the Intel Logo, the product and package type (P8048), the 4-digit Intel pattern number (WWWW), a date code (XXYY), and the customer part number (Z....Z). The customer part number is limited to a maximum of 9 digits or spaces.



1

P8048 MARKING EXAMPLE

FLOPPY DISK

Programming information may be sent on Intel Microcomputer Development System Floppy Disk. When using this input medium the floppy disk file name should be indicated in the Customer Part Number Section below. The type of floppy disk sent should also be indicated by checking one of the appropriate boxes:

☐ Single Density ☐ Double Density

CUSTOMER PART NUMBER

	Customer P/N (Please Fill-In)	Floppy Disk File Name (Please Fill-In)	Intel Pattern Number (Please Do Not Use)
1.			
2.			
3.			
4.			
5.			
6.			
7.			
8.			
9.			
10.			
11.			
12.			
13.			
14.			
15.			
16.			
17.			
18.			
19.			
20.			

INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 (408) 987-8080

CUSTOMER 8021 ROM ORDER FORM

I

Company _____ Phone No. _____

Company Contact _____ Date _____

P.O. No. _____ Package Type: □ Plastic □ Cerdip

For Intel Use Only

S No. _____

STD _____

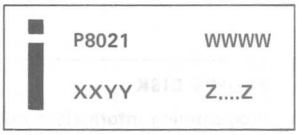
APP _____

DATE _____

All custom 8021 orders must be submitted on this form. Programming information should be sent in the form of computer punched cards or punched paper tape per the formats described in the Programming Instruction section of the Intel Data Catalog. 8748's may also be used to input programming information for the 8021. Additional forms are available from Intel.

MARKING

All devices will be marked as shown at the right figure. The marking will consist of the Intel Logo, the product and package type (P8021), the 4-digit Intel pattern number (WWWW), a date code (XXYY), and the customer part number (Z....Z). The customer part number is limited to a maximum of 9 digits or spaces.



P8021 MARKING EXAMPLE

CUSTOMER PART NUMBER

Customer P/N
(Please Fill-In)

└┴┴┴┴┴┴┴┴┴┘

Intel Pattern Number
(Please Do Not Use)

└┴┴┴┴┴┴┴┴┘

I/O Mask Options

Specify the desired connection for each I/O line on Port 0 and for the T1 input by marking only one box for each pin.

PIN		OPTION 1	OPTION 2
P00	4		
P01	5		
P02	6		
P03	7		
P04	8		
P05	9		
P06	10		
P07	11		
T1	13		

Port 0:

Option 1 deletes the pullup resistor on the I/O line providing true open drain outputs.

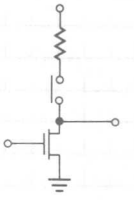
Option 2 includes the pullup resistor on the I/O line providing a quasi-bidirectional line.

T1:

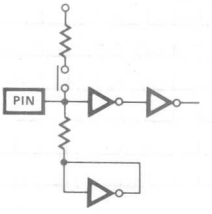
Option 1 deletes the pullup resistor for use as a zero cross detection input.

Option 2 includes the pullup resistor for use with an external switch or standard TTL.

OPTIONAL PULLUP RESISTOR



OPTIONAL PULLUP RESISTOR



INTEL CORPORATION, 3065 Bowers Avenue, Santa Clara, California 95051 (408) 987-8080

CUSTOMER 8022 ROM ORDER FORM

J

Company _____ Phone No. _____

Company Contact _____ Date _____

P.O. No. _____ Package Type: □ Plastic □ Cerdip

For Intel Use Only

S No. _____

STD _____

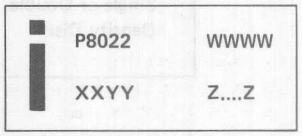
APP _____

DATE _____

All custom 8022 orders must be submitted on this form. Programming information should be sent in the form of computer punched cards or punched paper tape per the formats described in the Programming Instruction section of the Intel Data Catalog. 8755's may also used to input programming information for the 8022. Additional forms are available from Intel.

MARKING

All devices will be marked as shown at the right figure. The marking will consist of the Intel Logo, the product and package type (P8022), the 4-digit Intel pattern number (WWWW), a date code (XXYY), and the customer part number (Z....Z). The customer part number is limited to a maximum of 9 digits or spaces.



P8022 MARKING EXAMPLE

CUSTOMER PART NUMBER

Customer P/N
(Please Fill-In)

Intel Pattern Number
(Please Do Not Use)

I/O Mask Options

Specify the desired connection for each I/O line on Port 0 and for the T1 input by marking only one box for each pin.

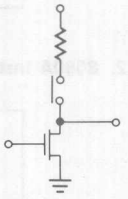
PIN		OPTION 1	OPTION 2
P00	10		
P01	11		
P02	12		
P03	13		
P04	14		
P05	15		
P06	16		
P07	17		
T1	19		

Port 0:

Option 1 deletes the pullup resistor on the I/O line providing true open drain outputs.

Option 2 includes the pullup resistor on the I/O line providing a quasi-bidirectional line.

OPTIONAL PULLUP RESISTOR

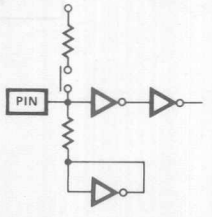


T1:

Option 1 deletes the pullup resistor for use as a zero cross detection input.

Option 2 includes the pullup resistor for use with an external switch or standard TTL.

OPTIONAL PULLUP RESISTOR



I. PROM AND ROM INPUT FORMATS

A. Acceptible Formats

Intel can accept programming and masking information for PROMs, EPROMs, or ROMs in the form of floppy disk, punched paper tape, a master device from which to copy, or computer punched cards. The allowable formats are given in Table 1. The preferred formats for the paper tape and computer card input media are the Intel Intellec Hex and BPNF since these formats are defined to allow detection of errors.

It is desirable that two, preferably different, input media for each customer code be sent so Intel can perform a code verification to detect any errors between the two inputs. This procedure, if followed, can avoid errors due to a mispunched tape/card or sending a defective or improper master device.

All orders must be accompanied by a customer PROM/ROM order form. A copy of the form is contained in this section and additional copies are available from Intel Literature Department, 3065 Bowers Ave., Santa Clara, California 95051.

Table I. Acceptable Formats

Floppy Disk	Paper Tape	Computer Card	Master Device
• Intel Microcomputer Development System Single or Double Density Disk	• Intellec Hex • BPNF • Hex	• Intellec Hex • PN	• Same Density and Pin Compatible to Device which is to be Programmed.

A1. Logic Levels

All data field for Intel's EPROMs/PROMs/ROMs are positive logic. The only exceptions are the 4001 and 4308 ROMs which use negative logic. For the 4001/4308, an "0" is a high output and a "1" is a low output. Consequently, because the BPNP format specifies the voltage level at the output of the device, it is necessary to input an "0" and "1" in the 4001/4308 instruction code as a "P" and "N" respectively. However, for the Hex format, the 4001/4308 input should be specified according to the instruction code logic state, i.e., a "1" or "0." The below example shows the corresponding input for 4001 instruction codes. For comparison, the input for an 8080A is also given as an example.

1. 4001 Instruction Code

4001 Instruction Mnemonic	4001 Instruction Code	Intellec Hex Or Non-Intellec Hex Input	BPNF Input
NOP	0000 0000	00	BPPPPPPPPF
WRM	1110 0000	E0	BNNNPPPPPF

2. 8080A Instruction Code

Instruction Mnemonic	Instruction Code	Intellec Hex Or Non-Intellec Hex Input	BPNF Input
JMP	1100 0011	C3	BPPNNNNPPF
Push D	1101 0101	D5	BPPNPNPNPF

B. Paper Tape Format

The paper tape which should be used is 1" wide paper using 7 or 8-bit ASCII code (such as a Model 33 ASR Teletype produces). The three paper tape formats which should be sent are described in Sections B1 through B3.

B1. Intellec Hex Paper Tape Format

In the Intel Intellec Hex Format, a data field can contain either 8 or 4-bit data. *Two* ASCII hexadecimal characters must be used to represent *both* 8 and 4-bit data. In the case of 4-bit data, only one of the characters is meaningful and must be specified on the Intel PROM/ROM Order Form.

Preceding the first data field and following the last data field there must be a leader/trailer length of at least 25 null characters. Comments (except for a colon) may be placed on the tape leader.

The format described below is readily generated by the Intel Intellec Microcomputer Development System or by systems programmed by the user.

1. RECORD MARK FIELD: Frame 0

 The ASCII code for a colon (:) is used to signal the start of a record.

2. RECORD LENGTH FIELD: Frames 1 and 2

 The number of data bytes in the record is representated by two ASCII hexadecimal digits in this field. The high-order digit is in frame 1. The maximum number of data bytes in a record is 255 (FF in hexadecimal). An end-of-file record contains two ASCII zeros in this field.

3. LOAD ADDRESS FIELD: Frames 3–6

 The four ASCII hexadecimal digits in frames 3–6 give the address at which the data is loaded. The high-order digit is in frame 3, the lower-order digit in frame 6. The first data byte is stored in the location indicated by the load address; successive bytes are stored in successive memory locations. This field in an end-of-file record contains zeros or the starting address of the program.

4. RECORD TYPE FIELD: Frames 7 and 8

 The two ASCII hexadecimal digits in this field specify the record type. The high-order digit is in frame 7. All data records are type 0; end-of-file records are type 1. Other possible values for this field are reserved for future expansion.

5. DATA FIELD: Frames 9 to 9+2*(record length)–1

 A data byte is represented by two frames containing the ASCII characters 0–9 or A–F, which represent a hexadecimal value between 0 and FF (0 and 255 decimal). The high-order digit is in the first frame of each pair. If the data is 4-bit, then either the high or low-order digit represents the data and the other digit of the pair may be any ASCII hexadecimal digit. There are no data bytes in an end-of-file record.

B3. Non-Intellec Hex Paper Tape Format

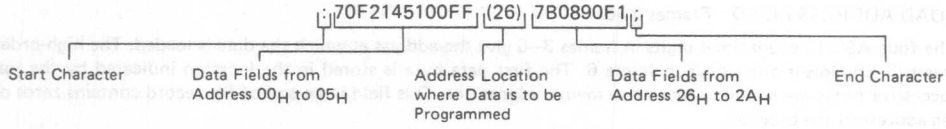
For the non-Intellec Hex Format, a data field can contain either 8 or 4-bit data. *Two* ASCII hexadecimal characters must be used to represent *both* 8 and 4-bit data. In the case of 4-bit data, only one of the characters is meaningful and must be specified on the Intel PROM/ROM Order Form.

Parity is allowed; however, it is not checked. Preceding the first data field and following the last data field there must be a leader/trailer length of at least 25 null characters or rubout punches.

The format requirements are as follows:

1. The start of the first data field is indicated by a colon. After the last data field, a semicolon must be punched to indicate the end. All data fields are to be punched in consecutive order, starting with data field 00_H (all addresses low).
2. *Two* hex characters must be used to represent the data field of both N word x 8-bit and N word x 4-bit devices. For an 8-bit data field, the high order data is represented by the left justified character of the pair. Either character of the pair may be used to represent the word field of a N word x 4-bit device, however, it must be consistent throughout the word field. The other character may be any hex character.

 A field of "don't care" data is allowed. Data after a field of "don't care" will be programmed starting at an address location enclosed in parentheses. In the following example, data is entered in addresses 00_H to 05_H, followed with "don't care" from addresses 06_H to 25_H, data being entered again starting at address location 26_H, and followed with "don't care" data to the last address location.



3. The x character may be used to rubout any erroneous character(s). The # character may be used to rubout an entire line up to the previous carriage return.
4. Spaces are allowed only between *separate* word fields.
5. After each 72 characters, a carriage return followed by a line feed should be punched to allow a print-out of the tape.
6. Comments must be placed only between the tape leader and the start of the first data field.

C. Computer Punched Card Format

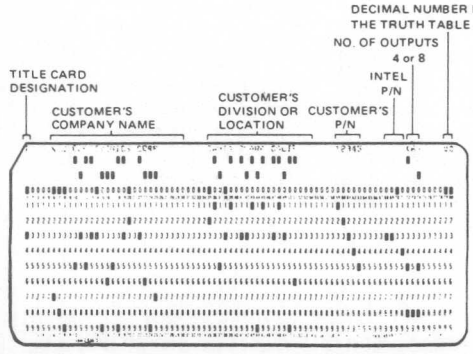
The following general format is applicable to the programming information sent on computer punched cards:

1. An 80 column Hollerith card (interpreted) punched on an IBM 026 or 029 keypunch should be submitted.
2. A single deck must consist of a Title Card followed by the data cards. There will be N/8 or N/14 data cards for N words x 8-bit and N words x 4-bit devices, respectively, in the PN format.

 For the Intellec Hex format, there will be N/32 data cards for both N words x 8-bit and N words x 4-bit devices, and one end of file card.

C1. Intellec Hex Computer Punched Card Format

Two hex characters must be used to represent data for both a N word x 8-bit and N word x 4-bit device. For the latter, only one of the characters is meaningful and must be specified on the Intel PROM/ROM Order Form. The entire data field for all bits must be punched even if it is "don't care".



Column	Data
1	Punch a T
2–3	Blank
4–28	Customer Company Name
29–30	Blank
31–50	Customer's Company Division or location
51–52	Blank
53–61	Customer Part Number
62–63	Blank
64–72	Punch the Intel 4-digit basic part number and in () the number of output bits; e.g., 2708 (8), 2316 (8), or 3605 (4)
73–74	Blank
75–76	Chip number for ROMs with programmable chip select inputs. If not applicable, leave blank.
77–78	Blank
79–80	Punch a 2-digit decimal number to indicate truth table number. The first truth table will be 00, second 01, third 02, etc.

a. N word x 8-bit device

Column	Data
1	Record mark: A colon is used to signal the start of a record.
2–3	Record length: This is the count of the actual data bytes in the record. Column 2 contains the high order digit of the count, Column 3 contains the low order digit. A record length of zero indicates end of file. All frames containing data will have a maximum record length of 10_{HEX} bytes (16 decimal).
4–7	Load address: The four characters starting addresses at which the following data will be loaded. The high order digit of the load address is in Column 4 and the low order digit is in Column 7. The first data byte is stored in the location indicated by the load address. Successive data bytes are stored in successive memory locations. ROMs containing more than 16 bytes of data will use two or more records or cards to transmit the data. Although the load address for the beginning record need not be 0000, each subsequent load address should be "10_H" (16 decimals) greater than the last.
8–9	Record type: A 2-digit code in this field specifies the type of this record. The high order digit of this code is located in Column 8. Currently, all data records are type 0. End-of-file records will be type 1; they are distinguished by a zero RECORD LENGTH field (see above). Other possible values for this field are reserved for future expansion.
10–73	Data
75–75	Checksum: Same as paper tape format.
76–78	Blank
79–80	Punch same 2-digit decimal number as in Title Card.

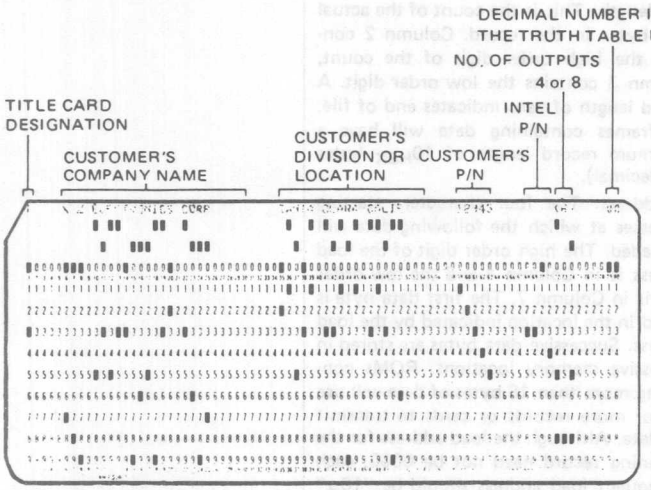
b. N word x 4-bit device

This format is identical to the previously documented 8-bit hexadecimal format with the following exceptions:

Column	Data
10–73	Each memory location is represented by two columns containing the characters 0–9, A–F. Since this is 4-bit data, the user must indicate which character of each pair is to be used as valid data. A single deck must be submitted without mixing first and second characters of the pair.

C2. PN Computer Punched Card Format

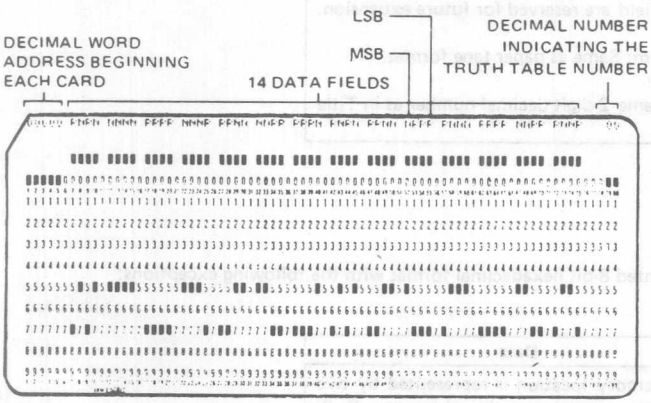
A word field consists of only P's and N's. A punched P will result in an output high level and a punched N in an output low level. The B and F characters, unlike the paper tape format, are illegal characters. The entire data field for all bits must be punched even if it is "don't care". The data field must begin in consecutive order, starting with address 0 (all addresses logically low).



Column	Data
1	Punch a T
2–3	Blank
4–28	Customer Company Name
29–30	Blank
31–50	Customer's Company Division or location
51–52	Blank
53–61	Customer Part Number
62–63	Blank
64–72	Punch the Intel 4-digit basic part number and in () the number of output bits; e.g., 2708 (8), 2316 (8), or 3605 (4)
73–74	Blank
75–76	Chip number for ROMs with programmable chip select inputs. If not applicable, leave blank.
77–78	Blank
79–80	Punch a 2-digit decimal number to indicate truth table number. The first truth table will be 00, second 01, third 02, etc.

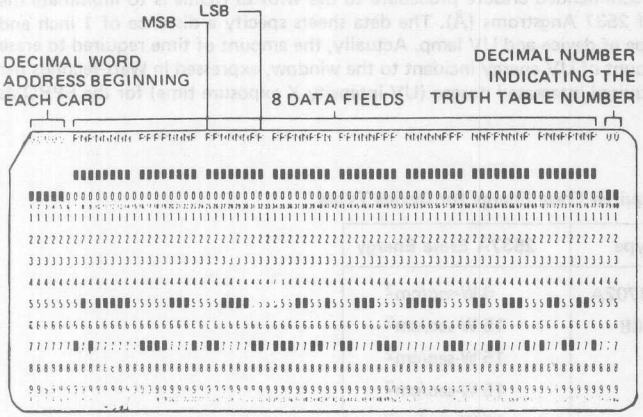
Title Card Format.

For a N words X 4-bit organization only, cards 2 and those following should be punched as shown. Each card specifies the 4-bit output of 14 words.



Column	Data
1-5	Punch the 5 digit decimal equivalent of the binary coded location which begins each card. The address is right justified, i.e., ØØØØØ, ØØØ14, ØØØ28, etc.
6	Blank
7-10	Data Field
11	Blank
12-15	Data Field
16	Blank
17-20	Data Field
21	Blank
22-25	Data Field
26	Blank
27-30	Data Field
31	Blank
32-35	Data Field
36	Blank
37-40	Data Field
41	Blank
42-45	Data Field
46	Blank
47-50	Data Field
51	Blank
52-55	Data Field
56	Blank
57-60	Data Field
61	Blank
62-65	Data Field
66	Blank
67-70	Data Field
71	Blank
72-75	Data Field
76-78	Blank
79-80	Punch same 2 digit decimal number as in title card.

For a N words X 8-bit organization only, cards 2 and those following should be punched as shown. Each card specifies the 8-bit output of 8 words.



Column	Data
1-5	Punch the 5 digit decimal equivalent of the binary coded location which begins each card. The address is right justified, i.e., 00000, 00008, 00016, etc.
6	Blank
7-14	Data Field
15	Blank
16-23	Data Field
24	Blank
25-32	Data Field
33	Blank
34-41	Data Field
42	Blank
43-50	Data Field
51	Blank
52-59	Data Field
60	Blank
61-68	Data Field
69	Blank
70-77	Data Field
78	Blank
79-80	Punch same 2 digit decimal number as in title card.

D. Custom PROM/ROM Order Forms

All orders for PROMs/ROMs which are to be electrically or mask programmed at Intel must be submitted with the order forms shown on the following pages. Additional forms are available from Intel Literature Department, 3065 Bowers Ave., Santa Clara, California 95051.

The order forms for the individual PROMs/ROMs are listed in Table II below.

Table II

PROM/ROM Part Number	Order Form Number
MOS EPROMs	A
8741, 8748, 8755	A
2316E, 8316A, 8316AL, 2616	B
2332, 2364	B
8041, 8048	C
2608, 8308	D
8355	E
Bipolar PROMs	F

II. MOS EPROMs

A. Erasure Procedure

As stated in the EPROM related data sheets, the recommended erasure procedure to use with EPROMs is to illuminate the window with a UV lamp which has a wavelength of 2537 Angstroms (Å). The data sheets specify a distance of 1 inch and erase times of 10—45 minutes, depending on the type of device and UV lamp. Actually, the amount of time required to erase a device can be concisely stated in terms of the amount of UV energy incident to the window, expressed in Watt-seconds per square centimeter (W-sec/cm^2). Table III lists the required integrated dosgae (UV intensity X exposure time) for the EPROMs currently in production by Intel.

Table III. Required Erase Energy for Device Types

Device Type	2537Å Erase Energy
1702A/4702A	6 W-sec/cm^2
2708/8708	15 W-sec/cm^2
2716	15 W-sec/cm^2
8748	15 W-sec/cm^2
8755	15 W-sec/cm^2

The erase energy expressed in Table III includes a guardband to ensure complete erasure of all bits. *It is not sufficient to monitor "first bit" erasure to determine erasure time, as some other bits in the array may not be erased.*

A1. UV Sources

There are several models of UV lamps that can be used to erase EPROMs (see Table IV). The model numbers in the table refer to lamps manufactured by Ultra Violet Products of San Gabriel, California. In addition, there are several other manufacturers, including Data I/O (Issaquah, Wash.), PROLOG (Monterey, Calif.). Prometrics (Chicago, Ill.), and Turner Designs (Mt. View, Calif.). The individual manufacturers should be consulted for detailed product descriptions. Also shown in the table are typical erase times for various combinations of Intel PROMs and lamp intensities.

Table IV.

Model	Power Rating	Minimum Erase Time for Indicated Dosage Without a Filter Over the Bulb	
		6 W-sec 1702A, 4702A	15 W-sec 2708, 8708, 8755 2716, 8748
R-52	13000 μW/cm^2	7.7 min	19.2 min
S-52	12000 μW/cm^2	8.3 min	20.7 min
S-68	12000 μW/cm^2	8.3 min	20.7 min
UVS-54	5700 μW/cm^2	17.5 min	43.8 min
UVS-11	5500 μW/cm^2	18.2 min	45.6 min

According to the manufacturers, the output of the UV lamp bulbs decrease with age. The output of the lamp should be verified periodically to ensure that adequate intensities are maintained. If this is not done, bits may be partially erased which will interfere with later programming and/or operation at high temperature.

For lamps other than those listed, the erase time can be determined by using a UV intensity meter, such as the Ultra Violet Products Model J-225. When a meter is used, the intensity should be measured at the same position (distance from the lamp) as the EPROMs to be erased. This will require careful positioning to insure that the sensor will receive the same amount of UV light that the window of the EPROM will receive.

The sensors used with most UV intensity meters show reduced output with constant exposure to UV light. Therefore, they should not be permanently placed inside the erasure enclosure, they should only be used for periodic measurements.

intel®

3065 Bowers Avenue
Santa Clara, California 95051
Tel: (408) 987-8080**
TWX: 910-338-0026
TELEX: 34-6372

U.S. AND CANADIAN DISTRIBUTORS

U.S. AND CANADIAN DISTRIBUTORS

ALABAMA

†*Hamilton/Avnet Electronics
805 Oser Drive NW
Huntsville 35805
Tel: (205) 533-1170

Pioneer
1207 Putman Drive NW
Huntsville 35805
Tel: (205) 837-9300

ARIZONA

†*Hamilton/Avnet Electronics
8155 North 21st Street
Phoenix 85021
Tel: (602) 275-7851

Liberty/Arizona
8155 N. 24th Avenue
Phoenix 85021
Tel: (602) 257-1272
TELEX: 910-951-4282

CALIFORNIA

†*Avnet Electronics
350 McCormick Avenue
Costa Mesa 92626
Tel: (714) 754-6111
Tel: (213) 558-2345

†*Hamilton/Avnet Electronics
575 E. Middlefield Road
Mountain View 94040
Tel: (415) 961-8600

†*Hamilton/Avnet Electronics
8917 Complex Drive
San Diego 92123
Tel: (714) 279-2421

†*Hamilton Electro Sales
10912 W. Washington Boulevard
Culver City 90230
Tel: (213) 558-2121

†Cramer/San Francisco
720 Palomar Avenue
Sunnyvale 94086
Tel: (408) 739-3011

Cramer/Los Angeles
17201 Daimler Street
Irvine 92714
Tel: (714) 979-3000

†Liberty Electronics
124 Maryland Street
El Segundo 90245
Tel: (213) 322-8100
Tel: (714) 638-7601
TWX: 910-348-7140

†Liberty/San Diego
8284 Mercury Court
San Diego 92111
Tel: (714) 565-9171
TELEX: 910-335-1590

†Elmar Electronics
2288 Charleston Road
Mountain View 94040
Tel: (415) 961-3611
TELEX: 910-379-6437

COLORADO

†Elmar/Denver
6777 E. 50th Avenue
Commerce City 80022
Tel: (303) 287-9611
TWX: 910-936-0770

†*Hamilton/Avnet Electronics
5921 No. Broadway
Denver 80216
Tel: (303) 545-1212

CONNECTICUT

†Cramer/Connecticut
35 Dodge Avenue
North Haven 06473
Tel: (203) 239-5641

†*Hamilton/Avnet Electronics
643 Danbury Road
Georgetown 06829
Tel: (203) 762-0361

Harvey Electronics
112 Main Street
Norwalk 06851
Tel: (203) 853-1515

FLORIDA

Cramer/E.W. Hollywood
4035 No. 29th Avenue
Hollywood 33020
Tel: (305) 921-7878

†*Hamilton/Avnet Electronics
6800 Northwest 20th Ave.
Ft. Lauderdale 33309
Tel: (305) 971-2900

Cramer/EW Orlando
345 No. Graham Ave.
Orlando 32814
Tel: (305) 894-1511

Pioneer
6220 S. Orange Blossom Trail
Suite 412
Orlando 32809
Tel: (305) 859-3600

GEORGIA

†Cramer
6456 Warren Drive
Norcross 30071
Tel: (404) 448-9050

**Note New Telephone Number

GEORGIA (cont.)

†*Hamilton/Avnet Electronics
6700 I 85, Access Road, #11
Norcross 30071
Tel: (404) 448-0800

ILLINOIS

†Cramer/Chicago
1911 So. Busse Rd.
Mt. Prospect 60056
Tel: (312) 593-8230

†*Hamilton/Avnet Electronics
3901 No. 25th Ave.
Schiller Park 60176
Tel: (312) 678-6310

Pioneer/Chicago
1551 Carmen Drive
Elk Grove Village 60006
Tel: (312) 437-9680

INDIANA

†Pioneer/Indiana
6408 Castleplace Drive
Indianapolis 46250
Tel: (317) 849-7300

Sheridan Sales
8790 Purdue Road
Indianapolis 46268
Tel: (317) 297-3146

KANSAS

†*Hamilton/Avnet Electronics
9219 Quivira Road
Overland Park 66215
Tel: (913) 888-8900

MARYLAND

†Cramer/EW Washington
16021 Industrial Drive
Gaithersburg 20760
Tel: (301) 948-0110

†*Hamilton Avnet
7235 Standard Drive
Hanover 21076
Tel: (301) 796-5000

†Pioneer/Washington
9100 Gaither Road
Gaithersburg 20760
Tel: (301) 948-0710
TWX: 710-828-0545

MASSACHUSETTS

†Cramer Electronics Inc.
85 Wells Avenue
Newton 02159
Tel: (617) 969-7700

†*Hamilton/Avnet Electronics
100 E. Commerce Way
Woburn 01801
Tel: (617) 933-8000

MICHIGAN

†Sheridan Sales Co.
24543 Indoplex Circle
Farmington Hills 48024
Tel: (313) 477-3800

†Pioneer/Michigan
13485 Stamford
Livonia 48150
Tel: (313) 525-1800

†*Hamilton/Avnet Electronics
32487 Schoolcraft Road
Livonia 48150
Tel: (313) 522-4700
TWX: 810-242-8775

MINNESOTA

†Industrial Components
5280 West 74th Street
Minneapolis 55435
Tel: (612) 831-2666

†Cramer/Bonn
7275 Bush Lake Road
Edina 55435
Tel: (612) 835-7811

†*Hamilton/Avnet Electronics
7683 Washington Avenue So.
Edina 55435
Tel: (612) 941-3801

MISSOURI

†*Hamilton/Avnet Electronics
396 Brookes Lane
Hazelwood 63042
Tel: (314) 731-1144

Sheridan Sales
110 S. Hwy. 67, Suite 10
Florissant 63031
Tel: (314) 837-5200

NEW JERSEY

Cramer/Pennsylvania, Inc.
12 Springdale Road
Cherry Hill Industrial Center
Cherry Hill 08034
Tel: (609) 424-5993
TWX: 710-896-0908

†*Hamilton/Avnet Electronics
218 Little Falls Road
Cedar Grove 07009
Tel: (201) 239-0800
TWX: 710-994-5787

Cramer/New Jersey
1 Cardinal Drive
Little Falls 07424
Tel: (201) 785-4300

NEW JERSEY (cont.)

†Harvey Electronics
389 Passaic Avenue
Fairfield 07006
Tel: (201) 227-1262

†*Hamilton/Avnet Electronics
113 Gaither Drive
East Gate Industrial Park
Mt. Laurel 08057
Tel: (609) 234-2133
TWX: 710-897-1405

NEW MEXICO

†*Hamilton/Avnet Electronics
2524 Baylor Drive, S.E.
Albuquerque 87119
Tel: (505) 765-1500

NEW YORK

†Cramer/Rochester
3000 Winton Road South
Rochester 14623
Tel: (716) 275-0300

†*Hamilton/Avnet Electronics
167 Clay Road
Rochester 14623
Tel: (716) 442-7820

Cramer/Syracuse
6716 Joy Road
East Syracuse 13057
Tel: (315) 437-6671

†*Hamilton/Avnet Electronics
6500 Joy Road
E. Syracuse 13057
Tel: (315) 437-2641

†Cramer/Long Island
129 Oser Avenue
Hauppauge, L.I. 11787
Tel: (516) 231-5600
TWX: 510-227-9863

†*Hamilton/Avnet Electronics
70 State Street
Westbury, L.I. 11590
Tel: (516) 333-5800
TWX: 510-222-8237

†Harvey Electronics
60 Crossways Park West
Woodbury 11797
Tel: (516) 921-8700

NORTH CAROLINA

†Cramer Electronics
938 Burke Street
Winston-Salem 27102
Tel: (919) 725-8711

Pioneer/Carolina
2906 Baltic Avenue
Greensboro 27406
Tel: (919) 273-4441
TWX: 510-925-1114

Hamilton/Avnet Electronics, Inc.
2803 Industrial Drive
Raleigh 27609
Tel: (919) 829-8030

OHIO

†Sheridan Sales Co.
2501 Neff Road
Dayton 45414
Tel: (513) 223-3332

†Cramer/Cleveland
5835 Harper Road
Cleveland 44139
Tel: (216) 248-8400

†*Hamilton/Avnet Electronics
954 Senate Drive
Dayton 45459
Tel: (513) 433-0610
TWX: 810-450-2531

†Pioneer/Dayton
1900 Troy Street
Dayton 45404
Tel: (513) 236-9900

†Sheridan Sales Co.
10 Knollcrest Drive
Cincinnati 45222
Tel: (513) 761-5432
TWX: 810-461-2670

†Pioneer/Cleveland
4800 E. 131st Street
Cleveland 44105
Tel: (216) 587-3600

†*Hamilton/Avnet Electronics
761 Beta Drive, Suite E
Cleveland 44143
Tel: (216) 461-1400

†Sheridan Sales Co.
23224 Commerce Park Road
Beachwood 44122
Tel: (216) 831-0130

OKLAHOMA

†Components Specialties, Inc.
7920 E. 40th Street
Tulsa 74145
Tel: (918) 664-2820

OREGON

†Almac/Stroum Electronics
4475 S.W. Scholls Ferry Rd.
Portland 97225
Tel: (503) 292-3534

PENNSYLVANIA

†Sheridan Sales Co.
1717 Penn Avenue, Suite 5009
Pittsburgh 15221
Tel: (412) 244-1640

Pioneer/Pittsburgh
560 Alpha Drive
Pittsburgh 15238
Tel: (412) 782-2300

PENNSYLVANIA (cont.)

Pioneer/Delaware
141 Gibraltar Road
Horsham 19044
Tel: (215) 674-4000
TWX: 510-665-6778

TENNESSEE

Sheridan Sales Co.
6900 Office Park Circle
Knoxville 37919
Tel: (615) 588-5386

TEXAS

Component Specialties Inc.
8330 Burnett Road, Suite 101
Austin 78758
Tel: (512) 459-3308

†Cramer Electronics
13740 Midway Road
Dallas 75240
Tel: (214) 661-9300

†*Hamilton/Avnet Electronics
4445 Sigma Road
Dallas 75240
Tel: (214) 661-8661

†*Hamilton/Avnet Electronics
3939 Ann Arbor
Houston 77063
Tel: (713) 780-1771

†Component Specialties, Inc.
10907 Shady Trail, Suite 101
Dallas 75220
Tel: (214) 357-6511

†Component Specialties, Inc.
8585 Commerce Park Drive, Suite 590
Houston 77036
Tel: (713) 771-7237

UTAH

†*Hamilton/Avnet Electronics
1585 West 2100 South
Salt Lake City, 84119
Tel: (801) 972-2800

WASHINGTON

†*Hamilton/Avnet Electronics
13407 Northrup Way
Bellevue 98005
Tel: (206) 746-8750

†Almac/Stroum Electronics
5811 Sixth Ave. South
Seattle 98108
Tel: (206) 763-2300

†Liberty Electronics
1750 132nd Avenue NE
Bellevue 98005
Tel: (206) 763-8200

WISCONSIN

†*Hamilton/Avnet
2975 Moorland Road
New Berlin 53151
Tel: (414) 784-4510

CANADA

ALBERTA

†L. A. Varah Ltd.
4742 14th Street N.E.
Calgary T2E 6LT
Tel: (403) 276-8818
Telex: 13 825 89 77

BRITISH COLUMBIA

†L.A. Varah Ltd.
2077 Alberta Street
Vancouver V5Y 1C4
Tel: (604) 873-3211
TWX: 610-929-1068
Telex: 04 53167

ONTARIO

L.A. Varah, Ltd.
505 Kenora Avenue
Hamilton L8E-3P2
Tel: (416) 561-9311
TELEX: 061-8349

†*Hamilton/Avnet Electronics
3688 Nashua Drive, Unit GH
Mississauga L4V IM5
Tel: (416) 677-7432
TWX: 610-492-8867

†*Hamilton/Avnet Electronics
1735 Courtwood Cresc.
Ottawa K2C 3J2
Tel: (613) 226-1700
TWX: 610 562-1906

†Zentronics
141 Catherine Street
Ottawa, Ontario K2P 1C3
Tel: (613) 238-6411

†Zentronics
99 Norfinch Dr.
Downsview, Ontario M3N 1W8
Tel: (416) 635-2822
Telex: 02-021694

QUEBEC

†*Hamilton/Avnet Electronics
2670 Paulus
St. Laurent H4S 1G2
Tel: (514) 331-6443
TWX: 610-421-3731

†Microcomputer System Technical Demonstrator Centers
*Microcomputer System Spares Order Point

HEXADECIMAL INSTRUCTION CODES

ACCUMULATOR

• ADD A,R_r	6*
• ADD A,@R0	60
R1	61
• ADD A,#data	03
• ADDC A,R_r	7*
• ADDC A,@R0	70
R1	71
• ADDC A,#data	13
ANL A,R_r	5*
ANL A,@R0	50
R1	51
ANL A,#data	53
ORL A,R_r	4*
ORL A,@R0	40
R1	41
ORL A,#data	43
XRL A,R_r	D*
XRL A,@R0	D0
R1	D1
XRL A,#data	D3
INC A	17
DEC A	07
CLR A	27
CPL A	37
RL A	E7
• RLC A	F7
RR A	77
• RRC A	67
• DA A	57
SWAP A	47

DATA MOVES

MOV A,R_r	F*
MOV A,@R0	F0
R1	F1
MOV A,#data	23
MOV R,A$_r$	A*
MOV @R0,A	A0
R1,A	A1
MOV R_r,#data	B*
MOV @R0,#data	B0
R1,#data	B1
XCH A,R_r	2*
XCH A,@R0	20
R1	21
XCHD A,@R0	30
R1	31
MOV A,PSW	C7
• MOV PSW,A	D7
MOVX A,@R0	80
R1	81
MOVX @R0,A	90
R1,A	91
MOVP3 A,@A	E3
MOVP A,@A	A3

REGISTER

INC R_r	1*
DEC R_r	C*
INC @R0	10
R1	11
DJNZ R_r, addr	E*

FLAGS

• CLR C	97
• CPL C	A7
CLR F0	85
CPL F0	95
CLR F1	A5
CPL F1	B5

BRANCH

JMP addr	†4
JMPP @A	B3
DJNZ R_r,addr	E*
JC addr	F6
JNC addr	E6
JZ addr	C6
JNZ addr	96
JT0 addr	36
JNT0 addr	26
JT1 addr	56
JNT1 addr	46
JF0 addr	B6
JF1 addr	76
JTF addr	16
JNI addr	86
JB0 addr	12
JB1 addr	32
JB2 addr	52
JB3 addr	72
JB4 addr	92
JB5 addr	B2
JB6 addr	D2
JB7 addr	F2

TIMER

MOV A,T	42
MOV T,A	62
STRT T	55
STRT CNT	45
STOP TCNT	65
EN TCNTI	25
DIS TCNTI	35

CONTROL

EN I	05
DIS I	15
SEL RB0	C5
SEL RB1	D5
SEL MB0	E5
SEL MB1	F5
ENT0 CLK	75

SUBROUTINE

CALL addr	†4
RET	83
RETR	93

NO OP

NOP	00

INPUT/OUTPUT‡

IN A,P1	09
OUTL P1,A	39
ANL P1,#data	99
ORL P1, #data	89
IN A, P2	0A
OUT L P2, A	3A
ANL P2, #data	9A
ORL P2,#data	8A
INS A, BUS	08
OUTL BUS, A	02
ANL BUS, #data	98
ORL BUS, #data	88
MOVD A,Pp	0‡
MOVD Pp,A	3‡
ANLD Pp,A	9‡
ORLD Pp,A	8‡

• = Carry Flag Affected
* = See Table 1
‡ = See Table 2
† = See Table 3